

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re PATENT APPLICATION of :
Yoshihiro SHONA :
Serial No.: [NEW] : Mail Stop Patent Application
Filed: January 20, 2004 : Attorney Docket No. OKI.635
For: METHOD FOR REWRITING DATA IN A MEMORY

CLAIM OF PRIORITY

U.S. Patent and Trademark Office
2011 South Clark Place
Customer Window, Mail Stop Patent Application
Crystal Plaza Two, Lobby, Room 1B03
Arlington, VA 22202

Sir:

Applicant, in the above-identified application, hereby claims the priority date under the International Convention of the following Japanese application:

Appln. No. 2003-274737 filed July 15, 2003

as acknowledged in the Declaration of the subject application.

A certified copy of said application is being submitted herewith.

Respectfully submitted,

VOLENTINE FRANCOS, PLLC



Adam C. Volentine
Registration No. 33,289

12200 Sunrise Valley Drive, Suite 150
Reston, Virginia 20191
Tel. (703) 715-0870
Fax. (703) 715-0877

Date: January 20, 2004

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 2003年 7月15日
Date of Application:

出願番号 特願2003-274737
Application Number:
[ST. 10/C]: [JP 2003-274737]

出願人 沖電気工業株式会社
Applicant(s):



2003年10月24日

特許庁長官
Commissioner,
Japan Patent Office

今井 康夫



出証番号 出証特2003-3088006

【書類名】 特許願
【整理番号】 KS001985
【提出日】 平成15年 7月15日
【あて先】 特許庁長官殿
【国際特許分類】 G11C 16/06
【発明者】
 【住所又は居所】 東京都港区虎ノ門1丁目7番12号 沖電気工業株式会社内
 【氏名】 正名 芳弘
【特許出願人】
 【識別番号】 000000295
 【氏名又は名称】 沖電気工業株式会社
【代理人】
 【識別番号】 100085198
 【弁理士】
 【氏名又は名称】 小林 久夫
 【電話番号】 03(3580)1936
【選任した代理人】
 【識別番号】 100061273
 【弁理士】
 【氏名又は名称】 佐々木 宗治
【選任した代理人】
 【識別番号】 100060737
 【弁理士】
 【氏名又は名称】 木村 三朗
【選任した代理人】
 【識別番号】 100070563
 【弁理士】
 【氏名又は名称】 大村 昇
【手数料の表示】
 【予納台帳番号】 044956
 【納付金額】 21,000円
【提出物件の目録】
 【物件名】 特許請求の範囲 1
 【物件名】 明細書 1
 【物件名】 図面 1
 【物件名】 要約書 1
 【包括委任状番号】 9001071
 【包括委任状番号】 9001072
 【包括委任状番号】 9001062
 【包括委任状番号】 9002063

【書類名】 特許請求の範囲**【請求項 1】**

複数のエリアに分割され、この各エリアの同一位置にそれぞれデータが書き込まれるセクタを有するメモリを備え、

書換データを書き込む際、その書換データと 1 番目エリアの書換対象位置のデータとの排他的論理和をとり、さらに、この排他的論理和のデータと 2 番目エリアの前記位置のデータとの排他的論理和をとる処理を順次繰り返し、その過程で当該エリアの前記位置のデータが初期値であったとき、そのエリアの前記位置にそれまでに求めた排他的論理和のデータを書き込むことを特徴とするメモリのデータ書き換え方法。

【請求項 2】

前記処理を最終エリアまで行った際に初期値がなかったときは、各エリアの同一位置毎に、1 番目エリアのデータと 2 番目エリアのデータとの排他的論理和をとり、さらに、この排他的論理和のデータと 3 番目エリアのデータとの排他的論理和をとる処理を順次繰り返し、この処理の繰り返しによって前記セクタ内の全データを読み出して書換対象位置のデータを書換データに変更し、その後、前記セクタ内の全データを消去して、その変更データを含む新たな全データをそのセクタの 1 番目エリアに書き込むことを特徴とする請求項 1 記載のメモリのデータ書き換え方法。

【請求項 3】

複数のエリアに分割され、この各エリアに複数のデータが書き込まれるセクタを有するメモリを備え、

書換データを書き込む際、その書換データと前記セクタの 1 番目エリアのデータとの排他的論理和をとり、さらに、この排他的論理和のデータと 2 番目エリアのデータとの排他的論理和をとる処理を順次繰り返し、その過程で当該エリアのデータが初期値であったとき、そのエリアにそれまでに求めた排他的論理和のデータを書き込むことを特徴とするメモリのデータ書き換え方法。

【請求項 4】

前記処理を最終エリアまで行った際に初期値がなかったときは、前記セクタ内の全エリアを消去した後に、前記書換データをそのセクタの 1 番目エリアに書き込むことを特徴とする請求項 3 記載のメモリのデータ書き換え方法。

【請求項 5】

複数のエリアに分割され、この各エリアにデータが書き込まれるセクタと、このセクタの各エリアに格納されたデータのエリア番号を指定する複数のポインタデータが書き込まれるポインタ用セクタとを有するメモリを備え、

書換データを書き込む際、前記ポインタ用セクタの 1 番目ポインタデータと 2 番目ポインタデータとの排他的論理和をとり、さらに、この排他的論理和のポインタデータと 3 番目ポインタデータとの排他的論理和をとる処理を順次繰り返してエリア番号を読み出し、このエリア番号に基づく前記セクタのエリアからデータを読み出して前記書換データと変更し、その後、前記エリア番号と前記セクタのエリア数と比較し、前記エリア番号が前記エリア数より小さいときは、そのエリア番号に「1」をインクリメントしてその新たなエリア番号のエリアに変更データを書き込み、また、前記エリア番号と前記エリア数とが同一のときは、前記セクタ内の全データを消去した後に変更データをそのセクタの 1 番目エリアに書き込んで、前記エリア番号を「1」とすることを特徴とするメモリのデータ書き換え方法。

【請求項 6】

前記ポインタ用セクタに、「1」をインクリメントしたエリア番号、又は「1」のエリア番号を書き込む際、そのエリア番号と前記ポインタ用セクタの 1 番目ポインタデータとの排他的論理和をとり、さらに、この排他的論理和のポインタデータと 2 番目ポインタデータとの排他的論理和をとる処理を順次繰り返し、その過程で当該ポインタデータが初期値であったとき、そのポインタデータにそれまでに求めた排他的論理和のポインタデータを上書きし、前記処理を最終まで行った際に初期値がなかったときは、前記ポインタ用セ

クタ内の全ポインタデータを消去した後に、前記エリア番号をそのポインタ用セクタの 1 番目に書き込むことを特徴とする請求項 5 に記載のメモリのデータ書き換え方法。

【請求項 7】

複数のエリアに分割され、この各エリアの同一位置にそれぞれデータが書き込まれる二つのセクタと、この二つのセクタの何れか一方を選択する複数のセクタデータが書き込まれるセクタ用セクタとを有するメモリを備え、

書換データを書き込む際、前記セクタ用セクタの 1 番目セクタデータと 2 番目ポインタデータとの排他的論理和をとり、さらに、この排他的論理和のセクタデータと 3 番目セクタデータとの排他的論理和をとる処理を順次繰り返してセクタ選択データを読み出し、この選択データに基づく一方のセクタの 1 番目エリアの書換対象位置のデータと前記書換データとの排他的論理和をとり、さらに、この排他的論理和のデータと 2 番目エリアの前記位置のデータとの排他的論理和をとる処理を順次繰り返し、その過程で当該エリアの前記位置のデータが初期値であったとき、そのエリアの前記位置にそれまでに求めた排他的論理和のデータを書き込むことを特徴とするメモリのデータ書き換え方法。

【請求項 8】

前記処理を最終エリアまで行った際に初期値がなかったときは、各エリアの同一位置毎に、1 番目エリアのデータと 2 番目エリアのデータとの排他的論理和をとり、さらに、この排他的論理和のデータと 3 番目エリアのデータとの排他的論理和をとる処理を順次繰り返す、この処理の繰り返しによって前記セクタ内の全データを読み出して書換対象位置のデータを書換データに変更し、もう一方のセクタの 1 番目エリアに、その変更データを含む新たな全データを書き込んで、前記セクタ選択データに「1」をインクリメントすることを特徴とする請求項 7 に記載のメモリのデータ書き換え方法。

【請求項 9】

前記セクタ選択データに「1」をインクリメントした際、そのセクタ選択データと前記セクタ用セクタの 1 番目セクタデータとの排他的論理和をとり、さらに、この排他的論理和のセクタデータと 2 番目セクタデータとの排他的論理和をとる処理を順次繰り返す、その過程で当該セクタデータが初期値であったとき、そのセクタデータにそれまでに求めた排他的論理和のセクタデータを上書きし、前記処理を最終まで行った際に初期値がなかったときは、前記セクタ用セクタ内の全セクタデータを消去した後に、前記セクタ選択データをそのポインタ用セクタの 1 番目に書き込むことを特徴とする請求項 8 に記載のメモリのデータ書き換え方法。

【請求項 10】

データが書き込まれる複数のエリアと、メモリ選択データ及びエリア指定データが書き込まれた管理エリアとが設けられたセクタを有する二つのメモリを備え、

書換データを書き込む際、二つのメモリに設けられた管理エリアのメモリ選択データに基づいて何れか一方のメモリを選択し、かつ、そのメモリの管理エリアのエリア指定データに基づくエリアに前記書換データを書き込むと共に、他方のメモリの複数のエリア内のデータを分割して消去し、一方のメモリの各エリアにデータを書き込んだときに他方のメモリの各エリア内のデータの消去が完了するようにし、二つのメモリに交互に書換データを書き込めるように前記二つの管理エリアの各データで誘導できるようにしたことを特徴とするメモリのデータ書き換え方法。

【請求項 11】

前記二つのメモリに加えてキャッシュメモリを備え、

その二つのメモリのうち一方のメモリに書換データを書き込んだ際に、前記キャッシュメモリに同じ書換データを書き込み、その一方のメモリに書き込んだデータを読み出す際には、前記キャッシュメモリから同じデータを読み出すことを特徴とする請求項 10 に記載のメモリのデータ書き換え方法。

【書類名】 明細書**【発明の名称】 メモリのデータ書き換え方法****【技術分野】****【0001】**

本発明は、例えばフラッシュメモリのデータの書き換え方法に関するものである。

【背景技術】**【0002】**

従来、フラッシュメモリ内のデータ書込領域が複数のデータブロックに分割されている場合において、所定のデータブロックに書換データを書き込むとき、そのデータブロック内のデータを消去し、その後に、そのデータブロックに書換データを書き込むようにしている（例えば、特許文献1参照）。

【0003】

【特許文献1】 特開平6-324945号公報（第3頁、図9、図10）

【発明の開示】**【発明が解決しようとする課題】****【0004】**

しかしながら、前述した従来のデータ書き換え方法では、コンタクト付きICカードやコンタクトレスICカードのように数mSで書き換え可能な不揮発性メモリを書き換える必要がある場合には、書き換え時間が長すぎて対応できないと言う欠点があった。

また、1バイト又は1ワード単位でのデータの書き換えを行う場合には、1セクタ（ページ）単位に消去を行ってから書き換えるようにしているので、書き換えの必要のない部分まで書き換えを行ってしまい、書き換え回数が増えてしまう欠点があり、メモリセルの劣化を早める恐れがあった。

また、セレクトトランジスタを設けてバイト単位またはワード単位の消去を可能にすれば、書き換え回数は少なくできるが、メモリセルにセレクトトランジスタを追加しなければならず、フラッシュメモリのサイズが大幅に大きくなるという課題があった。

【課題を解決するための手段】**【0005】**

本発明に係るメモリのデータ書き換え方法は、複数のエリアに分割され、この各エリアの同一位置にそれぞれデータが書き込まれるセクタを有するメモリを備え、書換データを書き込む際、その書換データと1番目エリアの書換対象位置のデータとの排他的論理和をとり、さらに、この排他的論理和のデータと2番目エリアの前記位置のデータとの排他的論理和をとる処理を順次繰り返し、その過程で当該エリアの前記位置のデータが初期値であったとき、そのエリアの前記位置にそれまでに求めた排他的論理和のデータを書き込むようにした。

【発明の効果】**【0006】**

以上のように本発明によれば、書換データを書き込む際、その書換データとメモリの1番目エリアの書換対象位置のデータとの排他的論理和をとり、さらに、この排他的論理和のデータと2番目エリアの前記位置のデータとの排他的論理和をとる処理を順次繰り返し、その過程で当該エリアの前記位置のデータが初期値であったとき、そのエリアの前記位置にそれまでに求めた排他的論理和のデータを書き込むようにしたので、従来のデータ書き換えと比べ、データの消去がない分、高速でデータを書き換えることができ、また、一つのセクタを複数のエリアに分割して使用しているので、同一セルの書き換え回数を減らすことが可能になり、メモリセルの劣化を防止でき、セレクトトランジスタを挿入しなくてもバイト単位やワード単位でデータの書き換えが可能になるので、メモリサイズを小さく抑えることができる。

【発明を実施するための最良の形態】**【0007】**

実施の形態1.

図1は本発明の実施の形態1に係るメモリのデータ書換方法の説明図である。なお、図中に示す実線の矢印はデータ書き換え時のデータの流れを示し、破線の矢印はデータ読み出し時のデータの流れを示す。

実施の形態1のメモリは、例えば、フラッシュメモリ10からなり、一括消去が可能な一定バイト数をもつ一つのセクタとして複数のセクタ1～セクタnに分割されている。その内、セクタ1は、データ書換用として、複数のエリア1～エリアnにそれぞれmバイトサイズで分割されている。このエリア1～エリアnは、前述したように一つのセクタ1であるため、一度の消去処理により一括消去が可能な構成となっており、一方、書き込みは1バイト単位で行うことができる構成になっている。なお、通常、フラッシュメモリ10の消去状態と書き込み状態は、メモリセルの方式や読み出し方法により異なるが、ここでは、消去状態を「0」、書込状態を「1」として説明する。

【0008】

フラッシュメモリ10以外のシステム構成は、図示していないが、システム全体を制御するCPU等からなる制御部、制御プログラムが格納されたROM等のプログラムメモリ、各種のデータを一時保管するためのRAM等のワークメモリ20等からなっている。このワークメモリ20は、データ書換時に使用するエリア番号書き込みのための変数領域jと、データ読出時に使用するエリア毎のデータ番号を書き込むための変数領域kと、データ読出時に使用するエリア番号書き込みのための変数領域Lと、書換データ及びこの書換データをXORしたXORデータを書き込むための変数領域WDと、各エリアに保存されたデータを書き込むための変数領域RDと、各エリアのデータをXORして得られる真のデータを一時的に保存するためのデータ書込領域21とを有している。

【0009】

次に、実施の形態1の動作について図2及び図3のフローチャートを用いて説明する。図2は実施の形態1におけるデータ書き換え時の動作を示すフローチャート、図3は実施の形態1におけるデータ読み出し時の動作を示すフローチャートである。なお、図2のフローチャートはデータD[i]を書き換えるときの一例で、まず最初に、各エリア1～nのD1[i]～Dn[i]が消去状態の「00h」になっているときの動作を説明し、次に、そのD1[i]～Dn[i]にデータが書き込まれているときの動作を説明する。

【0010】

制御部（図示せず）は、1バイトの書換データD[i]の入力を検知すると、まず、フラッシュメモリ10のセクタ1のエリア番号である「1」をワークメモリ20の変数領域jに書き込み、前記の書換データD[i]をワークメモリ20の変数領域WDに書き込む（Step1）。次に、エリアjのi番目のデータ、この場合は、エリア1のi番目のデータD1[i]をワークメモリ20の変数領域RDに書き込む（Step2）。そして、変数領域WD、RDにそれぞれ書き込んだ書換データD[i]とデータD1[i]のXOR（排他的論理和）をとり、その結果のXORデータを変数領域WD内の書換データD[i]に上書きする（Step3）。その後、ワークメモリ20の変数領域RDに書き込んだエリア1のデータD1[i]が「00h」かどうかを判別し（Step4）、そのデータD1[i]が「00h」でないときは、変数領域jに「1」をインクリメントして「2」とするが（Step6）、変数領域RDに書き込んだエリア1のデータD1[i]が「00h」のときは、変数領域WDに書き込んだXORデータをエリア1のデータD1[i]に上書きし（Step5）、前述した一連の動作を終了する。この場合は、エリア1のD1[i]に書き込まれていたデータが「00h」であるため、そのD1[i]には、入力された真の書換データD[i]が書き込まれる。

【0011】

この後に、2回目の書換データD[i]が入力されたときは、フラッシュメモリ10のセクタ1のエリアを指定する「1」をワークメモリ20の変数領域jに書き込み、書換データD[i]をワークメモリ20の変数領域WDに書き込む（Step1）。次に、エリア1のi番目のデータD1[i]をワークメモリ20の変数領域RDに書き込む（Step2）。そして、変数領域WD、RDにそれぞれ書き込んだ書換データD[i]とデータD1[i]

】のXORをとり、その結果のXORデータを変数領域WD内のデータD[i]に上書きする(Step3)。その後、ワークメモリ20の変数領域RDに書き込んだエリア1のデータD1[i]が「00h」かどうかを判別する(Step4)。この場合は、前述したように、エリア1のデータD1[i]が「00h」でないので、変数領域jに「1」をインクリメントして「2」とし(Step6)、その変数領域jの値がセクタ1のエリア番号の「n」を超えたかどうかを判別する(Step7)。変数領域jの値が「n」を超えていたときはセクタ1内の全データD[0]～D[m]の読み出し処理に入るが(Step8)、この時点では、変数領域jの値が「2」で「n」以下であるため、次のエリア2のi番目のデータD2[i]を読み出し、変数領域RDのデータD1[i]に上書きする(Step2)。

【0012】

次いで、そのデータD2[i]と変数領域WDに書き込んだ前回のXORデータをさらにXORし、その結果の新たなXORデータを変数領域WD内のXORデータに上書きする(Step3)。その後は、前述したように変数領域RDに書き込んだエリア2のデータD2[i]が「00h」かどうかを判別し(Step4)、この場合は、エリア2のデータD2[i]が「00h」であるため、変数領域WD内の新たなXORデータをエリア2のデータD2[i]に上書きし(Step5)、データ書き換え処理を終了する。

【0013】

また、3回目の書換データD[i]が入力されたときは、前述したようにセクタ1のエリアを指定する「1」をワークメモリ20の変数領域jに書き込み、書換データD[i]をワークメモリ20の変数領域WDに書き込む(Step1)。次に、エリア1のi番目のデータD1[i]をワークメモリ20の変数領域RDに書き込む(Step2)。そして、変数領域WD、RDにそれぞれ書き込んだ書換データD[i]とデータD1[i]のXORをとり、その結果のXORデータを変数領域WDの書換データD[i]に上書きする(Step3)。その後、ワークメモリ20の変数領域RDに書き込んだエリア1のデータD1[i]が「00h」かどうかを判別する(Step4)。そのエリア1のデータD1[i]は「00h」でないので、変数領域jに「1」をインクリメントして「2」とし(Step6)、その変数領域jの値がセクタ1のエリア番号の「n」を超えたかどうかを判別する(Step7)。変数領域jの値が「2」で「n」以下であるため、エリア番号が「2」であるエリア2のi番目のデータD2[i]を読み出し、変数領域RDのデータD1[i]に上書きする(Step2)。

【0014】

次いで、そのデータD2[i]と変数領域WDに書き込んだ前回のXORデータをさらにXORし、その結果の新たなXORデータを変数領域WD内のXORデータに上書きする(Step3)。その後は、変数領域RDに書き込んだエリア2のデータD2[i]が「00h」かどうかを判別し(Step4)、このときは、エリア2のデータD2[i]が「00h」でないため、変数領域jに「1」をインクリメントして「3」とし(Step6)、その変数領域jの値がセクタ1のエリア番号の「n」を超えたかどうかを判別する(Step7)。変数領域jの値が「3」で「n」以下であるため、エリア番号が「3」であるエリア3のi番目のデータD3[i]を読み出し、変数領域RDのデータD2[i]に上書きする(Step2)。

【0015】

その後は、そのデータD3[i]と変数領域WDに書き込んだXORデータをXORし、その結果のXORデータを変数領域WD内のXORデータに上書きする(Step3)。次に、変数領域RDに書き込んだエリア3のデータD3[i]が「00h」かどうかを判別し(Step4)、この時点では、エリア3のデータD3[i]が「00h」であるため、変数領域WD内のXORデータをエリア3のデータD3[i]に上書きし(Step5)、データ書き換え処理を終了する。前述したStep1～Step7までの動作によるデータの書き換えはn回まで行うことができる。

【0016】

次に、各エリア1～nのD1[i]～Dn[i]に「00h」以外のデータが書き込ま

れているときの動作を説明する。

この場合は、 n 回後の書換データ $D[i]$ が入力されたときで、その書換データ $D[i]$ をセクタ 1 のエリア 1 ~ n の各データ $D1[i] \sim Dn[i]$ と繰り返し XOR した後に、Step 7 において、変数領域 j の値が「 n 」を超えたかどうかを判別する。この場合は、エリア n を超えるので、フラッシュメモリ 10 のセクタ 1 に書き込まれているデータ $D1[0] \sim Dn[m]$ から真のデータ $D[0] \sim D[m]$ を読み出す処理に入る (Step 8)。この読み出し処理については図 3 を用いて後述する。Step 8 において読み出した真のデータ $D[0] \sim D[m]$ をワークメモリ 20 のデータ書込領域 21 に一時的に保存し、ワークメモリ 20 に保存した真のデータ $D[0] \sim D[m]$ のうちデータ $D[i]$ に書換データ $D[i]$ を上書きして変更し (Step 9)、次いで、フラッシュメモリ 10 のセクタ 1 に書き込まれている全データ $D1[0] \sim Dn[m]$ を消去し (Step 10)、そして、データ $D[i]$ の変更が終了した真のデータ $D[0] \sim D[m]$ をワークメモリ 20 から読み出して、フラッシュメモリ 10 のセクタ 1 のエリア 1 に書き込み (Step 11)、書換データ $D[i]$ の書き換え処理を終了する。

【0017】

次に、図 3 に示すフローチャートに基づいてデータの読み出し時の動作を説明する。

制御部 (図示せず) は、図 2 に示す Step 8 で真のデータ $D1[0] \sim Dn[m]$ の読み出し処理に入ると、まず、セクタ 1 の各エリア 1 ~ n にそれぞれ書き込まれたデータの配列順である「0」をワークメモリ 20 の変数領域 k に書き込む (Step 21)。次に、フラッシュメモリ 10 のセクタ 1 のエリア番号の「1」をワークメモリ 20 の変数領域 L に書き込むと共に、ワークメモリ 20 のデータ書込領域 21 の $D[k]$ 、この場合は、 $D[0]$ に消去状態の「00h」を書き込んでデータ $D[0]$ とする (Step 22)。

【0018】

その後、セクタ 1 のエリア L の k 番目のデータ $DL[k]$ 、この時点では、エリア 1 の 1 番目のデータ $D1[0]$ を読み出し、データ書込領域 21 のデータ $D[0]$ と XOR をとり、その結果の XOR データをデータ書込領域 21 のデータ $D[0]$ の「00h」に上書きする (Step 23)。そして、変数領域 L に「1」をインクリメントして「2」とし (Step 24)、その値がセクタ 1 のエリア番号の「 n 」を超えたかどうかを判別する (Step 25)。変数領域 L の値が「 n 」を超えていた場合は、Step 23 ~ Step 25 の動作の繰り返しにより、真のデータ $D[0]$ がデータ書込領域 21 に書き込まれていると判断して、次のデータ列を読み出すために Step 26 に進むが、この時点では、変数領域 L の値が「2」で「 n 」を超えていないので Step 23 に戻る。

【0019】

この時、エリア 2 の 1 番目のデータ $D2[0]$ を読み出し、データ書込領域 21 に書き込んだ XOR データ $D[0]$ と XOR をとり、その結果の新たな XOR データをそのデータ書込領域 21 の XOR データ $D[0]$ に上書きする (Step 23)。そして、変数領域 L に「1」をインクリメントして「3」とし (Step 24)、その値がセクタ 1 のエリア番号の「 n 」を超えたかどうかの判別に入る (Step 25)。Step 23 ~ Step 25 の動作は、前述したように各エリア 1 ~ n の 1 番目のデータ $D1[0] \sim Dn[0]$ から真のデータ $D[0]$ を読み出すまで繰り返し行い、変数領域 L の値が「 n 」を超えたときに、変数領域 k に「1」をインクリメントして「1」とし (Step 26)、その値が「 m 」を超えたかどうかを判別する (Step 27)。変数領域 k の値が「 m 」を超えていた場合は、セクタ 1 の各エリア 1 ~ n に書き込まれている全データ $D1[0] \sim Dn[m]$ から真のデータ $D[0] \sim D[m]$ を読み出したと判断して、この処理を終了するが、変数領域 k の値が「1」であるため、Step 22 に戻って変数領域 L を再び「1」に設定し、次いで、ワークメモリ 20 のデータ書込領域 21 の $D[k]$ 、この場合は、 $D[1]$ に消去状態の「00h」を書き込んでデータ $D[1]$ とする。

【0020】

そして、エリア 1 の 2 番目のデータ $D1[1]$ を読み出し、データ書込領域 21 のデータ $D[1]$ と XOR をとって、その結果の XOR データをデータ書込領域 21 のデータ D

[1] の「00h」に上書きし (Step 23)、変数領域 L に「1」をインクリメントして「2」とし (Step 24)、その値がセクタ 1 のエリア番号の「n」を超えたかどうかを判別する (Step 25)。この時点では、変数領域 L の値が「2」で「n」を超えていないので Step 23 に戻って、Step 23 ~ Step 25 までの動作をセクタ 1 のエリアが「n」になるまで繰り返し行う。その後は、変数領域 k に「1」をインクリメントして「2」とし (Step 26)、その値が「m」を超えたかどうかを判別する (Step 27)。この時点では、変数領域 k の値が「2」であるため Step 22 に戻って、各エリア 1 ~ n の m 番目のデータ D1[m] ~ Dn[m] から真のデータ D[m] を読み出すまで、前述した一連の動作を繰り返し行う (Step 22 ~ Step 27)。この Step 22 ~ Step 27 の繰り返しにより、ワークメモリ 20 のデータ書込領域 21 に真のデータ D[0] ~ D[m] を書き込んだとき、このデータの読み出し処理を終了する。

【0021】

ここで、フラッシュメモリ 10 のセクタ 1 の各エリア 1 ~ n が消去状態 (00h) のときにデータを書き換えていった場合のデータの状態を具体的に図 4 に基づいて説明する。図 4 はデータを 3 回書き換えたときのデータの書換状態を示す図である。

各エリア 1 ~ n のデータが全て「00h」のときに (a) 参照)、1 番目の D[0] に書換データの「55h」を書き込む場合、まず、この「55h」をワークメモリ 20 の変数領域 WD に書き込み、次いで、エリア 1 の 1 番目の D1[0] の「00h」を読み出してワークメモリ 20 の変数領域 RD に書き込む。その後、変数領域 WD、RD にそれぞれ書き込んだ「55h」と「00h」の XOR をとり、その結果の「55h」を変数領域 WD の「55h」に上書きする。そして、ワークメモリ 20 の変数領域 RD に書き込んだデータが「00h」かどうかを判別し、この場合、そのデータが「00h」であるため、変数領域 WD に書き込んだ「55h」をエリア 1 の D1[0] の「00h」に上書きする (b) 参照)。

【0022】

次に、「55h」のデータを「AAh」に書き換える場合は、前述したように、この「AAh」を変数領域 WD に書き込み、次いで、エリア 1 の 1 番目の D1[0] の「55h」を読み出して変数領域 RD に書き込む。その後、変数領域 WD、RD にそれぞれ書き込んだ「AAh」と「55h」の XOR をとり、その結果の「FFh」を変数領域 WD の「AAh」に上書きする。そして、変数領域 RD に書き込んだデータが「00h」かどうかを判別し、この場合は、そのデータが「55h」であるため、エリア 2 の 1 番目の D2[0] にアクセスして「00h」を読み出して、変数領域 RD の「55h」に上書きし、この上書きした「00h」と変数領域 WD の「FFh」の XOR をとり、その結果の「FFh」を変数領域 WD に上書きする。その後、変数領域 RD に上書きしたエリア 2 の 1 番目のデータが「00h」かどうかを判別し、この場合、そのデータが「00h」であるため、変数領域 WD に書き込んだ「FFh」をエリア 2 の D2[0] の「00h」に上書きする (c) 参照)。また、3 回目の書き換えとして今度は「20h」に書き換える場合は、「55h」XOR「FFh」XOR「20h」=「8Ah」をエリア 3 の D3[0] の「00h」に上書きすることになる (d) 参照)。

【0023】

なお、図 3 (d) の状態からデータを読み出す場合は、まず、ワークメモリ 20 のデータ書換領域 21 の D[0] に「00h」を設定し、次いで、この「00h」とエリア 1 の「55h」の XOR をとり、さらに、この XOR 値とエリア 2 の「FFh」の XOR をとり、最後に、この XOR 値とエリア 3 の「8Ah」を XOR とることにより、3 回目に書き換えたデータ「20h」が読み出される。

【0024】

以上のように実施の形態 1 によれば、フラッシュメモリ 10 の一つのセクタ 1 を m バイトサイズで複数個のエリア 1 ~ n に分割し、書換データを書き込む際には、各エリア 1 ~ n のデータと繰り返し XOR をとりながら「00h」のエリアに保存するようにしたので、従来のデータ書き換えと比べ、データの消去がない分、高速でデータを書き換えること

ができる。

【0025】

また、前述したように一つのセクタ1を複数のエリア1～nに分割して使用しているので、同一セルの書き換え回数を減らすことが可能になり、メモリセルの劣化を防止でき、セレクトトランジスタを挿入しなくてもバイト単位やワード単位でデータの書き換えが可能になるので、メモリサイズを小さく抑えることができる。

【0026】

また、各エリア1～nにそれぞれ書き込まれているデータは直接書き込まれたものではなくXORデータであるため、悪意をもった第三者がフラッシュメモリ10のデータを解析しにくくなり、セキュリティの向上が期待できる。

【0027】

さらに、本実施の形態1によれば、書換データを書き込む際、複数のエリア1～nに空きエリアがない場合、各エリア1～nのデータを消去することになるが、この消去をn回に1回のみであるため、特定の領域のみでデータの書き換えを要する例えばICカード等に適用できる。さらに、コンタクトレスICカードの乗車券への使用の場合は、改札での処理は高速に書き換える必要があるが、精算機での処理は遅い書き換えでも良いので、セクタ1のエリア数を十分に大きくして精算時に消去を含むデータの書き換えを行うようにすれば乗車券にも適用できる。

【0028】

実施の形態2.

図5は本発明の実施の形態2に係るメモリのデータ書換方法の説明図である。

実施の形態2は、図中に示すようにフラッシュメモリ10のセクタをデータの数に応じて用意し、一つのセクタを例えば1バイトのデータ格納用に使用したものである。つまり、データD[0]～D[m]を書き込むようにした場合、m+1個のセクタ0～mを用意し、各セクタ0～mをそれぞれnバイトで構成している。

【0029】

ワークメモリ20は、データ書換時に使用するセクタ毎のエリア番号を書き込むための変数領域jと、データ読出時に使用するセクタ番号を書き込むための変数領域kと、データ読出時に使用するセクタ毎のエリア番号を書き込むための変数領域Lと、書換データを保持するための変数領域WD'と、書換データ及びこの書換データのXORデータを書き込むための変数領域WDと、各セクタ0～mのエリアに保存されたデータD1[0]～Dn[m]を書き込むための変数領域RDと、各セクタ0～mのデータをXORして得られる真のデータD[0]～D[m]を一時的に保存するためのデータ書込領域21とを有している。

【0030】

次に、実施の形態2におけるデータ書き換え時の動作を図6のフローチャートを用いて説明する。なお、図2で説明した実施の形態1と同じStep No. (Step2～Step7)には同一のStep No. を付し詳細な動作説明を割愛する。

制御部(図示せず)は、1バイトの書換データD[i]の入力を検知すると、まず、その書換データD[i]をワークメモリ20の変数領域WDに書き込み、セクタiのエリアの順番を示す「1」をワークメモリ20の変数領域jに書き込むと共に、変数領域WDに書き込んだ書換データD[i]を変数領域WD'に書き込む(Step1')。

【0031】

その後は、実施の形態1で述べたように、セクタiのエリア番号jがnを超えるまでに「00h」のデータDj[i]を確認したときは、そのエリアにXORしたデータを変数領域WDから読み込んで書き込み、また、「00h」のデータDj[i]がなくエリア番号jがnを超えたときは(Step2～Step7)、セクタiに書き込まれている全データD1[i]～Dn[i]を消去する(Step10)。そして、変数領域WD'から書換データD[i]を読み出して、セクタiの先頭エリアに書き込み(Step11')、書換データD[i]の書き換え処理を終了する。

【0032】

なお、各セクタ 0～mに書き込まれている全データ D [0]～D [m] の読み出し処理については、基本的には実施の形態 1（図 3 参照）と同じであるが、唯一異なる点は、読み出しデータ D [0]～D [m] が各セクタのデータの XOR を取ったものになっている点であり、実施の形態 2 では、前述した如くデータの数と同数のセクタを必要とする点である。

【0033】

以上のように実施の形態 2 によれば、1 バイトのデータに対し n エリア（n バイト）を有するセクタを設け、書き換えの度に XOR したデータを各エリアに追記して行くようにしたので、実施の形態 1 と同様に、メモリセルの劣化を防止でき、セレクトトランジスタを挿入しなくてもバイト単位やワード単位でデータの書き換えが可能になり、セキュリティの向上が期待できる。

また、書換データに対するセクタ内のエリアが XOR データでいっぱいになっていた場合に、全データ D [0]～D [m] を読み出すことなく、該当するセクタ内のデータのみを消去して書換データを書き込むようにしているので、実施の形態 1 と比べ処理が簡略化でき、処理時間を短くできる。

【0034】

さらに、一つのセクタで 1 バイトのデータを書き込むようにしているので、書き換えのターゲットであるデータ以外のデータを消去するようなことがなくなり、このため、他のデータが書き込み不良によりデータ化けを起こすような心配がないという効果がある。

【0035】

実施の形態 3.

図 7 は本発明の実施の形態 3 に係るメモリのデータ書換方法の説明図である。なお、図 1 及び図 5 で説明した実施の形態 1、2 と同一又は相当部分には同じ符号を付し説明を省略する。

実施の形態 3 におけるフラッシュメモリ 10 は、データ書込用のセクタ 1 と、例えば 1 バイトのポインタデータ P（以下、単に「ポインタ P」という）を格納するためのセクタ 2 とを有している。セクタ 1 は、実施の形態 1 と同様に m バイトサイズで分割された複数のエリア 1～n からなり、セクタ 2 は n バイトで構成されている。前記のポインタ P は、各データが書き込まれているエリア番号を示している。

【0036】

ワークメモリ 20 は、実施の形態 1 と同じ書込領域を備えている上に、ポインタ P 書換時に使用するセクタ 2 のエリア番号を書き込むための変数領域 j' と、ポインタ P 読出時に使用するセクタ 2 のエリア番号を書き込むための変数領域 L' と、書換ポインタ P を保持するための変数領域 PWD' と、書換ポインタ P 及びこの書換ポインタ P の XOR データを書き込むための変数領域 PWD と、セクタ 2 の各エリアに保存されたポインタ P 1～P n を書き込むための変数領域 PRD と、セクタ 2 のポインタ P 1～P n を XOR して得られる真のポインタ P を一時的に保存するためのデータ書込領域 PD とが設けられている。

【0037】

次に、実施の形態 3 の動作について図 8～図 11 のフローチャートを用いて説明する。図 8 は実施の形態 3 におけるデータ書き換え時の動作を示すフローチャート、図 9 は実施の形態 3 におけるデータ読み出し時の動作を示すフローチャート、図 10 は実施の形態 3 におけるポインタ P 読み出し時の動作を示すフローチャート、図 11 は実施の形態 3 におけるポインタ P 書き換え時の動作を示すフローチャートである。なお、図 2 及び図 3 並びに図 6 と同じ Step No.（Step 1～Step 7 と Step 10, 11 並びに Step 21～Step 27）については、動作が同じであるため、詳細な動作説明を省略する。

【0038】

制御部（図示せず）は、例えば 1 バイトの書換データ D [i] の入力を検知すると、その書換データ D [i] をワークメモリ 20 の変数領域 WD に格納し、セクタ 2 からポイン

タ P を読み出すための処理に入る (Step 3 1)。この読み出し処理は、図 1 0 に示すように、まず、ワークメモリ 2 0 の変数領域 L' にエリア番号を示す「1」を書き込み、次いで、変数領域 P D に消去状態の「0 0 h」を書き込む (Step 2 2)。その後、セクタ 2 の 1 番目のエリアに書き込まれたデータ P 1 と変数領域 P D の「0 0 h」の X O R をとり、変数領域 P D の「0 0 h」に上書きする (Step 2 3)。そして、変数領域 L' に「1」をインクリメントして「2」とし (Step 2 4)、その値が「n」を超えたかどうかを判別する (Step 2 5)。この時点では、変数領域 L' の値が「2」であるため Step 2 2 に戻って、前述した動作を繰り返す。この一連の動作を繰り返すうちに変数領域 L' の値が「n」を超えると、ポインタ P の読み出し処理を終了する。この読み出し処理によって、真のポインタ P が変数領域 P D に書き込まれる。

【0 0 3 9】

この読み出し処理が終了したときは、図 8 に示す Step 3 2 において、読み出したポインタ P とセクタ 1 のエリア数「n」とを比較する。P = n ならエリア n までデータが書き込まれていると判断して Step 3 3 に進むが、ポインタ P がセクタ 1 のエリア数「n」よりも少ないときは、D_j [i] の何れかに X O R データを書き込むエリアがあると判断して、実施の形態 1 と同様の Step 1 ~ Step 6 の処理を行う。この処理を実行した際に、Step 4 においてセクタ 1 から変数領域 R D に書き込んだデータが「0 0 h」と判断した場合は、X O R した書換データ W D をその「0 0 h」の D_j [i] に書き込み (Step 5)、書き換え処理を終了する。また、変数領域 R D 内のデータが「0 0 h」でないときは、変数領域 j (エリア数) に「1」をインクリメントし (Step 6)、その値が「P」を超えたかどうかを判別する (Step 7')。変数領域 j の値が「P」以下のときは Step 2 に戻って動作を繰り返し、変数領域 j の値が「P」を超えたときは、その「P」を超えたエリア j にデータが書き込まれていないと判断してそのエリア j の D_j [i] に、X O R した書換データ W D を書き込む (Step 3 8)。そして、読み出したポインタ P に「1」をインクリメントし (Step 3 9)、セクタ 2 にそのポインタ数の「P」を書き込む処理に入る (Step 4 0)。

【0 0 4 0】

一方、Step 3 2 において、P = n と判断したときは、前述したようにエリア n までデータが書き込まれていると判断して、セクタ 1 から全データ D [0] ~ D [m] を読み出し (Step 3 3)、ワークメモリ 2 0 のデータ書込領域 2 1 に格納する。この読み出し処理については、図 9 を用いて後述する。その後は、データ書込領域 2 1 に格納した全データ D [0] ~ D [m] のうち D [i] に書換データ D [i] を上書きして変更する (Step 3 4)。その後、セクタ 1 に書き込まれた全 X O R データ D 1 [0] ~ D n [m] を消去し (Step 3 5)、セクタ 1 のエリア 1 に前記の全データ D [0] ~ D [m] を書き込む (Step 3 6)。そして、ポインタ P を「1」とし (Step 3 7)、前記と同様に「P」の書き込み処理に入る (Step 4 0)。

【0 0 4 1】

この処理は、図 6 で説明した実施の形態 2 の書き換え処理と同じであるが、書き換え処理時に使用する変数領域が異なるため、図 1 1 に示すフローチャートに基づいて説明する。

まず、図 8 の Step 3 7 又は Step 3 9 の処理によるポインタ P をワークメモリ 2 0 の変数領域 P W D に書き込み、次いで、セクタ 2 のエリアの順番を示す「1」をワークメモリ 2 0 の変数領域 j' に書き込むと共に、前記のポインタ P をワークメモリ 2 0 の変数領域 P W D' に書き込む (Step 1')。

【0 0 4 2】

そして、セクタ 2 の 1 番目のエリア P 1 に書き込まれたポインタ P 1 を変数領域 P R D に書き込み (Step 2)、変数領域 P W D、P R D にそれぞれ書き込んだポインタ P とポインタ P 1 の X O R (排他的論理和) をとり、その結果の X O R ポインタを変数領域 P W D 内のポインタ P に上書きする (Step 3)。その後は、ワークメモリ 2 0 の変数領域 P R D に書き込んだポインタ P 1 が「0 0 h」かどうかを判別し (Step 4)、そのポインタ P 1 が「0 0 h」でないときは、変数領域 j' に「1」をインクリメントして「2」とするが

(Step6)、変数領域 P R D に書き込んだポインタ P 1 が「00h」のときは、変数領域 P W D に書き込んだ X O R ポインタをセクタ 2 の 1 番目のエリア P 1 に上書きし (Step 5)、前述した一連の動作を終了する。

【0043】

また、セクタ 2 の各エリアに「00h」のポインタ P がいない状態でエリア番号 j' が n を超えたときは (Step 2 ~ Step 7)、セクタ 2 に書き込まれている全ポインタ P 1 ~ P n を消去する (Step 10)。そして、変数領域 P W D' から書き換えのポインタ P を読み出して、セクタ 2 の先頭エリア P 1 に書き込み (Step 11')、ポインタ P の書き換え処理を終了する。

【0044】

一方、セクタ 1 から真のデータ D [0] ~ D [m] を読み出す場合は、図 9 に示すように、まず最初、図 10 のフローチャートに基づいてポインタ P を読み出す (Step 31)。このポインタ P の読み出し処理が終了した後は、図 3 で説明した実施の形態 1 とほぼ同様の処理を実行して、セクタ 1 から真のデータ D [0] ~ D [m] を読み出す (Step 21 ~ Step 27)。実施の形態 1 では、Step 25 で「L」と「n」とを比較しているのに対し、本実施の形態 3 では、Step 25' において、エリア番号を示す「L」とデータが書き込まれたエリアの番号を示すポインタ数の「P」 (Step 31) とを比較している点異なるだけである。この比較処理によって、セクタ 1 のどのエリアまでデータが書き込まれているかがわかり、エリアからデータを読み込んで X O R する回数を最少にすることができる。

【0045】

以上のように実施の形態 3 によれば、フラッシュメモリ 10 に、実施の形態 1 と同じデータ書込用のセクタ 1 と、セクタ 1 のエリアのポインタデータ P を書き込むためのセクタ 2 とを用意したので、実施の形態 1 と同様に、メモリセルの劣化を防止でき、セレクトトランジスタを挿入しなくてもバイト単位やワード単位でデータの書き換えが可能になり、しかも、セクタ毎に書き込むデータが X O R 値であるため、セキュリティの向上が期待できる。

【0046】

また、ポインタデータ P を読み出すことにより、セクタ 1 のエリアがいつばいで追記できない場合に素早く判断することが可能になり、データの消去と書込み処理が速やかに行うことができ、さらに、読み出し時にもポインタデータ P から使用されているエリアが分かるので、データ読み出しと X O R 処理の無駄をなくすことができ、読み出し速度を速くできる。

【0047】

実施の形態 4.

図 12 は本発明の実施の形態 4 に係るメモリのデータ書換方法の説明図である。なお、図 1 及び図 7 で説明した実施の形態 1、3 と同一又は相当部分には同じ符号を付し説明を省略する。

実施の形態 4 におけるフラッシュメモリ 10 は、実施の形態 3 と同様に、データ書込用のセクタ 1 と、データが書き込まれているエリア番号を示す 1 バイトのポインタデータ P (以下、単に「ポインタ P」という) を格納するためのセクタ 2 とを有している。セクタ 1 は、実施の形態 1 と同様に m バイトサイズで分割された複数のエリア 1 ~ n からなり、この各エリア 1 ~ n には、X O R されたデータを書き込むのではなく、直接、真のデータを書き込むようになっている。セクタ 2 は、実施の形態 3 で述べたように、n バイトで構成され、X O R されたポインタ P が格納されるようになっている。

【0048】

次に、実施の形態 4 の動作について図 13 及び図 14 のフローチャートを用いて説明する。図 13 は実施の形態 4 におけるデータ書き換え時の動作を示すフローチャート、図 14 は実施の形態 4 におけるデータ読み出し時の動作を示すフローチャートである。なお、実施の形態 3 で用いた図 8 ~ 図 11 と同じ Step No. については、動作が同じであるため、詳細な動作説明を省略する。

【0049】

制御部（図示せず）は、例えば1バイトの書換データ $D_p[i]$ の入力を検知すると、そのデータを一時的にワークメモリ20の変数領域WDに格納し、次いで、ポインタP（エリア番号）に応じたエリアからデータ $D_p[0] \sim D_p[m]$ を読み出して、ワークメモリ20のデータ書込領域21に格納する（Step51）。この読み出し処理については図14に基づいて後述する。そして、データ書込領域21に格納したデータ $D_p[0] \sim D_p[m]$ のうちデータ $D_p[i]$ に書換データ $D_p[i]$ を上書きして変更する（Step52）。その後、セクタ2からポインタPを読み出し（Step31）、セクタ1のエリア番号の「n」と比較する（Step32）。なお、ポインタPの読み出し処理については、図10で説明した実施の形態3と同様であるため説明は割愛する。

【0050】

ポインタPと「n」と比較した際に $P = n$ と確認したときは、セクタ1のエリアnまでデータでいっぱいだと判断して、セクタ1に書き込まれている全データを消去し（Step35）、データ書込領域21に格納したデータ $D_p[0] \sim D_p[m]$ をセクタ1のエリア1に書き込む（Step36）。そして、ポインタPを「1」にし、セクタ2への書き込み処理に入る（Step40）。この処理については、図11で説明した実施の形態3と同様であるため説明は割愛する。

【0051】

一方、ポインタPと「n」が等しくないときは、空きエリアがあると判断して、そのポインタPに「1」をインクリメントし（Step53）、その値と同じエリアpに前記のデータ $D_p[0] \sim D_p[m]$ を書き込んで（Step54）、「1」をインクリメントしたポインタPをセクタ2に書き込むための処理を行い（Step40）、データ書き換えの動作を終了する。

【0052】

次に、図14に基づいてデータ読み出し時の動作を説明する。まず、図10で説明した実施の形態3と同じ方法でセクタ2からポインタPを読み出す（Step31）。その後、各エリア1～nに配列されているデータの順番を示す変数領域kを「0」にし（Step21）、次いで、ポインタPが示すエリアpの1番目のデータ $D_p[0]$ を読み出してワークメモリ20のデータ書込領域21に書き込む（Step62）。そして、変数領域kに「1」をインクリメントして「2」とし（Step26）、バイト数の「m」を超えたかどうかを判別する（Step27）。超えていたときはこの動作を終了するが、「k」が「m」を超えていないときは、同一エリアp内にデータがあると判断して、2番目のデータ $D_p[2]$ を読み出してワークメモリ20のデータ書込領域21に書き込む（Step62）。この一連の動作を同一のエリアp内に書き込まれているデータが読み出されるまで繰り返し行う（Step62～Step27）。

【0053】

以上のように実施の形態4においては、実施の形態3のようにデータを書き換えるときや読み出すときにXORすることなく、セクタ2のポインタPの示すエリアに直接データを書き込んだり読み出すようにしたので、メモリセルの劣化の防止や、セレクトトランジスタを挿入することなくバイト単位やワード単位でデータの書き換えが可能になり、また、読み出しの際にXORして行く必要がないので、データの書き換えと読み出し処理が実施の形態3と比べ速くなるという効果がある。

【0054】

実施の形態5.

図15は本発明の実施の形態5に係るメモリのデータ書換方法の説明図である。なお、図1及び図7で説明した実施の形態1、3と同一又は相当部分には同じ符号を付し説明を省略する。

実施の形態5は、フラッシュメモリ10に、データを書き込むための二つのセクタ0、1と、例えば1バイトのセクタデータS（以下、単に「セクタS」という）を書き込むためのセクタ2とを設けたものである。前記セクタSは、セクタ0又はセクタ1を選

択するためのデータで、例えば、その値が偶数のときはセクタ 0 が、奇数のときはセクタ 1 が選択される。

【0055】

ワークメモリ 20 には、実施の形態 1 と同様の各種変数領域及びデータ格納領域 21 が設けられている上に、セクタ S に応じて 1 又は 0 を書き込むための変数領域 q、q' と、セクタ S の書換時にセクタ 2 のエリア番号を書き込むための変数領域 j' と、セクタ S の読出時にセクタ 2 のエリア番号を書き込むための変数領域 l' と、書換セクタ S を保持するための変数領域 SWD' と、書換セクタ S 及びこの書換セクタ S の XOR データを書き込むための変数領域 SWD と、セクタ 2 の各エリアに保存されたセクタ S1 ~ Sn を書き込むための変数領域 SRD と、セクタ 2 のセクタ S1 ~ Sn を XOR して得られる真のセクタ S を一時的に保存するためのデータ書込領域 SD とが設けられている。

【0056】

次に、実施の形態 5 の動作について図 16 及び図 17 のフローチャートを用いて説明する。図 16 は実施の形態 5 におけるデータ書き換え時の動作を示すフローチャート、図 17 は実施の形態 5 におけるデータ読み出し時の動作を示すフローチャートである。なお、図中に示す () 内の Step No. は、図 2 及び図 3 で説明した実施の形態 1 と同じであるため、動作の説明を省略する。

【0057】

制御部 (図示せず) は、例えば 1 バイトの書換データ D[i] の入力を検知すると、その書換データ D[i] をワークメモリ 20 の変数領域 WD に格納し、セクタ 2 からセクタ S を読み出す処理を実行する (Step 61)。この読み出し処理については、図 10 で説明した実施の形態 3 と同じで、この図に示す PD を SD、PL' を SL' と置き代えて処理する。制御部は、この処理によってセクタ 2 から読み出した真のセクタ S を一旦変数領域 SD に格納し、そのセクタ S が偶数かどうかを判別する (Step 62)。セクタ S が偶数のときは、セクタ 0 を選択するために変数領域 q に「0」、変数領域 q' に「1」を書き込み、また、セクタ S が奇数のときは、セクタ 1 を選択するために変数領域 q に「1」、変数領域 q' に「0」を書き込む。

【0058】

その後は、図 2 で説明した実施の形態 1 と同様の動作を実行する (Step 1 ~ Step 7)。Step 62 においてセクタ 0 を選択した場合は、変数領域 q の「0」に基づいてセクタ 0 の各エリア 1 ~ n からデータ D01[i] ... を順に読み出して XOR して行き、途中で「00h」のデータを確認したときは、その位置に、XOR した書換データを書き込んで動作を終了する。一方、セクタ 0 の各エリア 1 ~ n に「00h」以外のデータが書き込まれていた場合は、そのセクタ 0 から真のデータ D[0] ~ D[m] を読み出してデータ書込領域 21 に保存し (Step 8)、かつ、その中のデータ D[i] に変数領域 WD 内の書換データ D[i] を上書きして変更する (Step 9)。そして、真のデータ D[0] ~ D[m] をもう一方のセクタ 1 のエリア 1 に書き込む (Step 65)。

【0059】

また、Step 62 においてセクタ 1 を選択した場合は、変数領域 q の「1」に基づいてセクタ 1 の各エリア 1 ~ n からデータ D11[i] ... を順に読み出して XOR して行き、途中で「00h」のデータを確認したときは、その位置に、XOR した書換データを書き込んで動作を終了する。一方、セクタ 1 の各エリア 1 ~ n に「00h」以外のデータが書き込まれていた場合は、そのセクタ 1 から真のデータ D[0] ~ D[m] を読み出してデータ書込領域 21 に保存し (Step 8)、かつ、その中のデータ D[i] に変数領域 WD 内の書換データ D[i] を上書きして変更する (Step 9)。そして、真のデータ D[0] ~ D[m] をもう一方のセクタ 0 のエリア 1 に書き込む (Step 65)。なお、Step 8 の動作は、図 3 で説明した実施の形態 1 と同様であり、Step 9 は、前述したように実施の形態 1 と同じである。

【0060】

セクタ0又はセクタ1へのデータ書込が終了すると、変数領域SDのセクタSの値に「1」をインクリメントし(Step66)、そのセクタSの書き込み処理の動作を実行する(Step67)。この書き込み処理については、図11で説明した実施の形態3と同じで、この図に示すPWDをSWD、PWD'をSWD'、Pj'をSj'、PRDをSRDとそれぞれ置き代えて処理する。この処理によって変数領域SWD'に書き込んだセクタSをセクタ2の先頭エリアに書き込む。そして、Step62で選択したセクタ0のデータD01[0]~D0n[m]、又はセクタ1のデータD11[0]~D1n[m]を消去し(Step68)、データの書き換え処理を終了する。このデータの消去を最後に行うようにしたのは、もう一方のセクタのエリア1にデータを書き込む前に消去すると、処理の途中で電源が落ちたりして中断した場合にデータが消去してしまうので、これを防止するためである。

【0061】

次に、図17に基づいてデータ読み出し時の動作を説明する。まず、図16で説明したようにセクタ2からセクタSを読み出し、ワークメモリ20の変数領域SDに保存する(Step61)。そして、そのセクタSが偶数かどうかを判別し(Step72)、セクタSが偶数のときは変数領域qに「0」を書き込み(Step73)、セクタSが奇数のときはその変数領域qに「1」を書き込む(Step74)。変数領域qに「0」を書き込んだ場合は、その変数に基づきセクタ0に書き込まれているデータD01[0]~D0n[m]を読み出してXORして行き、データ書込領域21に一旦保存し(Step21~Step27)、また、変数領域qに「1」を書き込んだ場合は、その変数に基づきセクタ1に書き込まれているデータD11[0]~D1n[m]を読み出してXORして行き、データ書込領域21に一旦保存し(Step21~Step27)、データの読み出し処理を終了する。この処理は、図3で説明した実施の形態1と同じである。

【0062】

以上のように実施の形態5によれば、データを書き込むための二つのセクタ0, 1を設け、さらに、セクタSを書き込むためのセクタ2を設け、このセクタSに基づいて二つのセクタ0, 1を交互に使用するようにしたので、実施の形態1と同様に、メモリセルの劣化を防止でき、セレクトトランジスタを挿入しなくてもバイト単位やワード単位でデータの書き換えが可能になり、しかも、セクタ毎に書き込むデータがXOR値であるため、セキュリティの向上が期待でき、また、書き換え途中で電源が落ちて処理が中断されても、データが消失することがないという効果が期待できる。

【0063】

実施の形態6.

図18は本発明の実施の形態6に係るメモリのデータ書換方法の説明図である。

実施の形態6は、図中の(a)に示すように、複数のセクタ01~n、11~nをそれぞれ有する2つのフラッシュメモリ0, 1(以下、単に「メモリ」という)を備え、この2つのメモリ0, 1の同一番目のセクタ01, 11をデータ格納エリアとして使用する構成になっている。各セクタ01, 11は、例えば同図(b)に示すように、所定のデータ長を格納可能な4つのエリア1~4と管理エリア5a, 5bとで構成されている。この管理エリア5a, 5bは、例えば同図(c)に示すように、2ビットのメモリセクタと4ビットのエリアセクタとから構成されている。なお、管理エリア5aはメモリ0側のセクタ01に、もう一方の管理エリア5bはメモリ1側のセクタ11に設けられているものとする。

【0064】

管理エリア5a, 5bのメモリセクタ内の2ビットの値は、メモリ0, 1の何れかを選択するためのもので、各値の組み合わせをキーとして選択するようになっている。例えば、メモリ0(m0)側のメモリセクタが「01」で、メモリ1(m1)側のメモリセクタが「X0」のときはメモリ0と判断して、変数jを「0」にし、また、メモリ0側のメモリセクタが「0X」で、メモリ1側のメモリセクタが「01」のときはメモリ1と判断して、変数jを「1」にする。各メモリセクタ内の値とメモリ番号は(図18

(e) 参照)、データとしてROM (図示せず) に格納されており、変数 j はワークメモリ (図示せず) の所定エリアに書き込むようになっている。

【0065】

管理エリア 5 a、5 b のエリアセクタ内の 4 ビットの値は、エリアを選択するためのもので、データ書き換え時と読み出し時に応じて異なっている。例えば、データ書き換え時において、管理エリア 5 a、5 b のメモリセクタからメモリ 0 と判断した場合、メモリ 0 の管理エリア 5 a のエリアセクタに書き込まれている値に応じてそのメモリ 0 のセクタ 0 1 のエリア番号を判別する。このエリア番号は変数 k として前記のワークメモリの所定エリアに書き込むようになり、エリアセクタ内の値とエリア番号は (図 18 (d) 参照)、前記と同様にデータとしてROM に格納されている。

【0066】

次に、実施の形態 6 の動作について図 19 及び図 20 を用いて説明する。図 19 は実施の形態 6 におけるデータ書き換え時の動作を示すフローチャート、図 20 はデータ書き換え処理に基づくセクタの状態を示す図である。なお、動作説明の便宜上、各セクタ 0 1、1 1 のそれぞれのエリア 1 ~ 5 が全て消去状態の「00h」になっているものとする (図 20 (a) 参照)。

【0067】

データの書き換えが 1 回目の場合は、まず、各セクタ 0 1、1 1 の管理エリア 5 a、5 b のメモリセクタからそれぞれ値を読み出し (Step 8 1)、メモリ番号の選択に入る。この時点では、前述したように各メモリセクタ内の値が共に「00」であるため、Step 8 2 においてメモリ 0 と判断して、変数 j を「0」にする (Step 8 3)。メモリ 0 を選択した場合は、セクタ 0 1 のエリアセクタ内の値を読み出し、データを書き込むエリア番号の判別に入る。この時は、その値が「0000」であるため、Step 9 2 においてエリア 1 と判断して、変数 k を「1」にする (Step 9 3)。

【0068】

以上の判別からメモリ 0 のセクタ 0 1 のエリア 1 にデータを書き込むと共に、もう一方のメモリ 1 のセクタ 1 1 に書き込まれているデータを 1/4 時間消去する (Step 10 2)。これは、そのセクタ 1 1 のエリア 1 内のデータを消去することで、そのエリア 1 にデータが書き込まれているか否かの判別を行うことなく消去処理を実行するようになっている。このデータの消去について、例えば、消去時間を 10 mS、書き込み時間を 20 μ S とした場合、データの消去時間を 2.5 mS で良く、書き込み速度は速いので、この消去の間に書き込むことは十分可能である。また、メモリを 2 つに分けた事により、ほぼ同時に書き込みと消去を可能にしている。

【0069】

データの書き込みが終了すると、セクタ 0 1 の管理エリア 5 a 内の各値を更新する (Step 10 3)。この処理も消去中の 2.5 mS の中で行い、更新には消去を伴わない「0」から「1」への変更のみとする。ここでは、管理エリア 5 a のメモリセクタの値「00」を「01」に変更し、エリアセクタの値「0000」を次の書き込みエリアを示す「0001」に変更する。これで 1 回目の書き換え処理が完了し、図 20 (b) に示すような状態となる。

【0070】

次に、データの書き換えが 2 回目の場合は、前記と同様に管理エリア 5 a、5 b のメモリセクタからそれぞれの値を読み出し (Step 8 1)、メモリ番号の選択に入る。この場合は、管理エリア 5 a のメモリセクタが「01」で、管理エリア 5 b のメモリセクタが「00」であるため、Step 8 4 においてメモリ 0 と判断して、変数 j を「0」にする (Step 8 5)。その後は、選択したメモリ 0 側のエリアセクタ内の値を読み出し、データを書き込むエリア番号の判別に入る。この時は、その値が「0001」であるため、Step 9 4 においてエリア 2 と判断して、変数 k を「2」にする (Step 9 5)。

【0071】

そして、選択したメモリ 0 のセクタ 0 1 のエリア 2 にデータを書き込むと共に、もう一

方のメモリ1のセクタ11に書き込まれているデータを1/4時間消去する(Step102)。そして、メモリ0の管理エリア5aのメモリセクタの値「01」を保持し、エリアセクタの値「0001」を「0011」に変更し(Step103)、2回目の書き換え処理が完了する。この時の状態は図20(c)である。

【0072】

3回目の場合は、Step81からStep84に進んでメモリ番号を示す変数jを「0」にし(Step85)、その後は、メモリ0側のエリアセクタ内の値が「0011」であるため、Step96からStep97に進んでエリア番号を示す変数kを「3」にする。そして、選択したメモリ0のセクタ01のエリア3にデータを書き込むと共に、前記と同様にメモリ1のセクタ11に書き込まれているデータの1/4を消去し(Step102)、メモリ0の管理エリア5a内の値を変更する。この場合は、メモリセクタの値「01」を保持し、エリアセクタの値「0011」を「0111」に変更し(Step103)、3回目の書き換え処理が完了する(図20(d)参照)。

【0073】

4回目の場合は、Step81→Step84→Step85に進んで変数jを「0」にし、さらに、メモリ0側のエリアセクタ内の値「0111」に基づいてStep98→Step99に進んで変数kを「4」にする。そして、この判別からメモリ0のセクタ01のエリア4にデータを書き込み、メモリ1のセクタ11に書き込まれている残り1/4時間データを消去し(Step102)、メモリ0の管理エリア5a内の値を変更する。この場合は、メモリセクタの値「01」を保持し、エリアセクタの値「0111」を「1111」に変更し(Step103)、4回目の書き換え処理が完了する(図20(e)参照)。これにより、メモリ0側の4つの全エリア1~4にデータが書込まれた状態となる。

【0074】

データの書き換えが5回目の場合は、管理エリア5a、5bのメモリセクタからそれぞれの値を読み出す(Step81)。この時は、図20(e)に示すように、管理エリア5aのメモリセクタが「01」で、管理エリア5bのメモリセクタが「00」であるため、Step84においてメモリ0と判断して、変数jを「0」にする(Step85)。その後は、選択したメモリ0側のエリアセクタ内の値を読み出し、データを書き込むエリアの番号の判別に入る。この時は、メモリ0側のエリアセクタの値が「1111」であるため、Step100において、Step85で設定した変数jの「0」に「1」を加算し、その値を2で除算したときのMOD(余り)を変数jとする。この場合、MODは「1」となるので、メモリ番号を示す変数jを「1」に変更し、エリア番号を示す変数kを「1」に設定する。

【0075】

この処理からメモリ1のセクタ11のエリア1にデータを書き込むと共に、メモリ0のセクタ01に書き込まれているデータを消去する(Step102)。この場合も消去時間が2.5mS(1/4時間消去)であるが、メモリセルにより消去時間にバラツキがあるので、メモリ0のセクタ01のデータは不定となり、図20(f)に示すような状態になる。メモリ1側のエリア1にデータを書き込んだ後は、前記と同様に、管理エリア5bのメモリセクタの値「00」を「01」に変更し、エリアセクタの値「0000」を次の書き込みエリアを示す「0001」に変更する(Step103)。これで5回目の書き換え処理が完了する(図20(f)参照)。

【0076】

これ以降にデータを書き換える場合は、メモリ1のセクタ11のエリア2から順に書き込んでいき、データの書き換えが8回目の場合は、そのセクタ11のエリア4に書き込み、管理エリア5bのメモリセクタの値「01」を保持し、エリアセクタの値を「1111」に変更する。この時、メモリ0のセクタ01は2.5mS×4回=10mSの消去を受けたことになり、完全に消去された状態「00h」になっている(図20(g)参照)。

【0077】

また、データの書き換えが9回目の場合は、図20 (g) に示すように、管理エリア5 aのメモリセクタが「00」で、管理エリア5 bのメモリセクタが「01」であるため、Step 88においてメモリ1と判断して、変数jを「1」にするが (Step 85)、そのメモリ1側のエリアセクタの値が「1111」であるため、Step 100において、Step 89で設定した変数jの「1」に「1」を加算し、その値を2で除算したときのMOD (余り) を変数jとする。この場合、MODは「0」となるので、メモリ番号を示す変数jを「0」に変更し、エリア番号を示す変数kを「1」に設定する。つまり、メモリ1からメモリ0に変更し、そのエリアの番号を「1」にする。

【0078】

そして、この処理からメモリ0のセクタ01のエリア1にデータを書き込み、メモリ1のセクタ11に書き込まれているデータを消去する (Step 102)。この場合も、メモリセルにより消去時間にバラツキがあるので、メモリ1のセクタ11のデータは不定となり、図20 (h) に示すような状態になる。その後は、前記と同様に、管理エリア5 aのメモリセクタの値を「10」に変更し、エリアセクタの値「0000」を「0001」に変更し (Step 103)、9回目の書き換え処理が完了する (図20 (h) 参照)。前述した管理エリア5 aのメモリセクタの変更は、メモリ1側の管理エリア5 bのメモリセクタの値を左に1回ローテーションさせてシフトした値である。その後は、同じように書き換え処理を行うとメモリ0のセクタ01がいっぱいになり、今度はメモリ1のセクタ11の消去が完了する (図20 (i) 参照)。

【0079】

これ以降にデータを書き換える場合は、前述したようにStep 100でメモリ0からメモリ1に切り換わり、データの書き換えが繰り返される (図20 (j) (k) 参照)。そして、メモリ1のセクタ11がいっぱいになったときは、メモリ0に切り換わって、再び図20 (b) からデータの書き換えが繰り返し行われる (同図 (b) ~ (k) 参照)。

【0080】

次に、データ読み出し時の動作を図21に示すフローチャートに基づいて説明する。

例えば、メモリ0、1の各セクタ01、11内が図20 (e) に示すような状態になっているときのデータを読み出す場合は、まず、各セクタ01、11の管理エリア5 a、5 bのメモリセクタからそれぞれ値を読み出し (Step 111)、メモリ番号の選択に入る。この時は、管理エリア5 aのメモリセクタの値が「01」で、管理エリア5 bのメモリセクタの値が「00」であるため、Step 114においてメモリ0と判断して、変数jを「0」にする (Step 115)。このメモリ0を選択した場合は、セクタ01のエリアセクタに書き込まれている値を読み出し、データを読み出すエリアの番号の判別に入る。この時は、その値が「1111」であるため、Step 130においてエリア4と判断して、変数kを「4」にする (Step 131)。以上の判別からメモリ0のセクタ01のエリア4に書き込まれている最新のデータを読み出し (Step 132)、前述した一連の動作を完了する。

【0081】

以上のように実施の形態6においては、2つのメモリ0、1 (フラッシュメモリ) を用意し、各セクタ01、11を所定のデータ長で複数のエリア1~4に分割すると共に管理エリア5 a、5 bをそれぞれ設け、一方のメモリのエリアに追記書込みすると同時に他方のメモリのセクタ消去を分割して行い、一方のメモリのセクタの各エリアにデータを書き込んだときに、他方のメモリのセクタの消去が完了するようにして、2つのメモリに交互にデータを書き込むようにしたので、データの書き換え時間を大幅に短縮できる。

【0082】

また、前述した実施の形態1~5では、通常の書き換え時間は短いですが、エリアがいっぱいになったときには、セクタ消去を行う必要があり、この時の書き換え処理時間は短くできなかったが、本実施の形態6では、データの書き込みと同時に分割消去を行っていくので、全てのデータの書き換えにおいて書き換え時間を短くできる。

【0083】

さらに、本実施の形態6では、セレクトトランジスタの付加なしに、バイトやワード単位でのデータの書き換えが可能であり、メモリセル構成をシンプルにでき、また、セクタ内の複数のエリアに順に書き込んでいくので、書き換え回数を少なくすることが可能でメモリセルの劣化を防止できる。

【0084】

実施の形態7.

図22は本発明の実施の形態7に係るメモリのデータ書換方法の説明図である。

実施の形態7は、複数のセクタ01～n、11～nをそれぞれ有する2つのフラッシュメモリ0、1と、キャッシュメモリ6とを備えたものである。その2つのフラッシュメモリ0、1のセクタ01、11には、前述した実施の形態6と同じ構成からなる4つのエリア1～4と管理エリア5a、5bとが設けられている。前記のキャッシュメモリ6は、フラッシュメモリ0、1と同じデータ容量を有し、各セクタ01、11に設けられた4つのエリア1～4にそれぞれ対応するエリア61～64が設けられている。

【0085】

次に動作について説明する。本実施の形態におけるデータの書き換え動作は、実施の形態6と殆ど同じであるが、フラッシュメモリ0又はフラッシュメモリ1の何れか一方の例えばエリア3にデータを書き込む際に、そのエリア3に対応するキャッシュメモリ6のエリア63にも同じデータを格納する。データを読み出すときは、フラッシュメモリ0、1から読み出すのではなく、キャッシュメモリ6に格納された最新のデータを読み出す。ただし、電源投入時には、実施の形態6に示す読み出し方法でフラッシュメモリ0又は1に書き込まれている最新のデータを読み出して、キャッシュメモリ6に格納する処理が必要になる。

【0086】

以上のように実施の形態7によれば、実施の形態6の構成にキャッシュメモリ6を追加し、フラッシュメモリ0又は1にデータの書き換えを行う際に、その最新のデータをキャッシュメモリ6に格納するようにしたので、実施の形態6と同等の効果が得られる上にデータの読み出し処理が簡略化され、読み出し速度を上げることができる。また、キャッシュメモリ6からデータを読み出すようにしているので、データ読み出し時の消費電力を低減できるという効果も期待できる。

【0087】

なお、前記の実施の形態1～7では、フラッシュメモリの書き換え処理と読み出し処理をCPU等の制御部と制御プログラムによるシフト処理として説明したが、それに限定されるものではなく、ハードウェアによって行っても良い。また、消去状態を「0」、書き込み状態を「1」として説明したが、これに限定されるものではない。ただし、実施の形態1～5において「1」を消去状態とした場合には、XOR追記の関係上nを奇数にするか、又はnが偶数の場合はXORする必要がある。

【0088】

さらに、実施の形態1～5についても、実施の形態7のようなキャッシュメモリ6を設けて、読み出し処理を速めるようにしても良い。また、実施の形態6では、1セクタ内のエリア数を4つの例で説明したが、これに限定されるものではなく、分割されて行われる消去時間の合計が消去に必要な時間になるようにエリア分割すれば良い。この方法を用いると、理論的には、書き込みに必要な時間まで書き換え時間を短縮できる。

【図面の簡単な説明】

【0089】

【図1】本発明の実施の形態1に係るメモリのデータ書換方法の説明図である。

【図2】実施の形態1におけるデータ書き換え時の動作を示すフローチャートである。

【図3】実施の形態1におけるデータ読み出し時の動作を示すフローチャートである。

【図4】データを3回書き換えたときのデータの書換状態を示す図である。

【図 5】本発明の実施の形態 2 に係るメモリのデータ書換方法の説明図である。

【図 6】実施の形態 2 におけるデータ書き換え時の動作を示すフローチャートである。

【図 7】本発明の実施の形態 3 に係るメモリのデータ書換方法の説明図である。

【図 8】実施の形態 3 におけるデータ書き換え時の動作を示すフローチャートである。

【図 9】実施の形態 3 におけるデータ読み出し時の動作を示すフローチャートである。

【図 1 0】実施の形態 3 におけるポインタ P 読み出し時の動作を示すフローチャートである。

【図 1 1】実施の形態 3 におけるポインタ P 書き換え時の動作を示すフローチャートである。

【図 1 2】本発明の実施の形態 4 に係るメモリのデータ書換方法の説明図である。

【図 1 3】実施の形態 4 におけるデータ書き換え時の動作を示すフローチャートである。

【図 1 4】実施の形態 4 におけるデータ読み出し時の動作を示すフローチャートである。

【図 1 5】本発明の実施の形態 5 に係るメモリのデータ書換方法の説明図である。

【図 1 6】実施の形態 5 におけるデータ書き換え時の動作を示すフローチャートである。

【図 1 7】実施の形態 5 におけるデータ読み出し時の動作を示すフローチャートである。

【図 1 8】本発明の実施の形態 6 に係るメモリのデータ書換方法の説明図である。

【図 1 9】実施の形態 6 におけるデータ書き換え時の動作を示すフローチャートである。

【図 2 0】データ書き換え処理に基づくセクタの状態を示す図である。

【図 2 1】実施の形態 6 におけるデータ読み出し時の動作を示すフローチャートである。

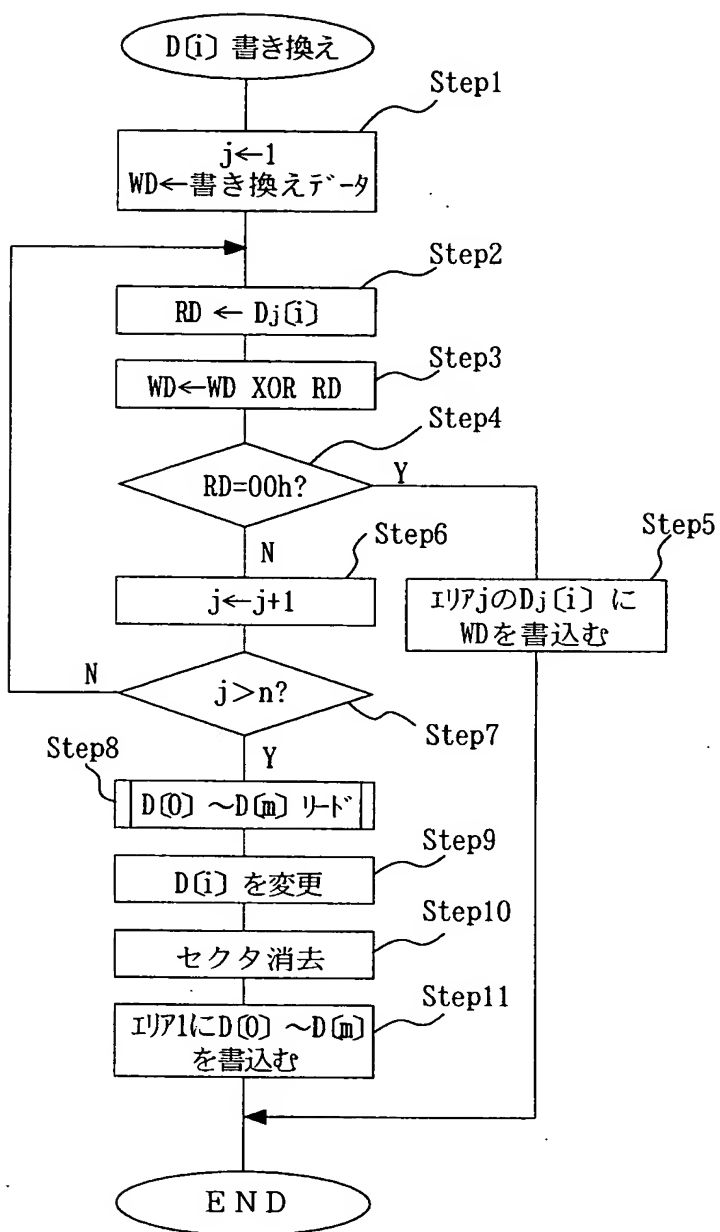
【図 2 2】本発明の実施の形態 7 に係るメモリのデータ書換方法の説明図である。

【符号の説明】

【 0 0 9 0 】

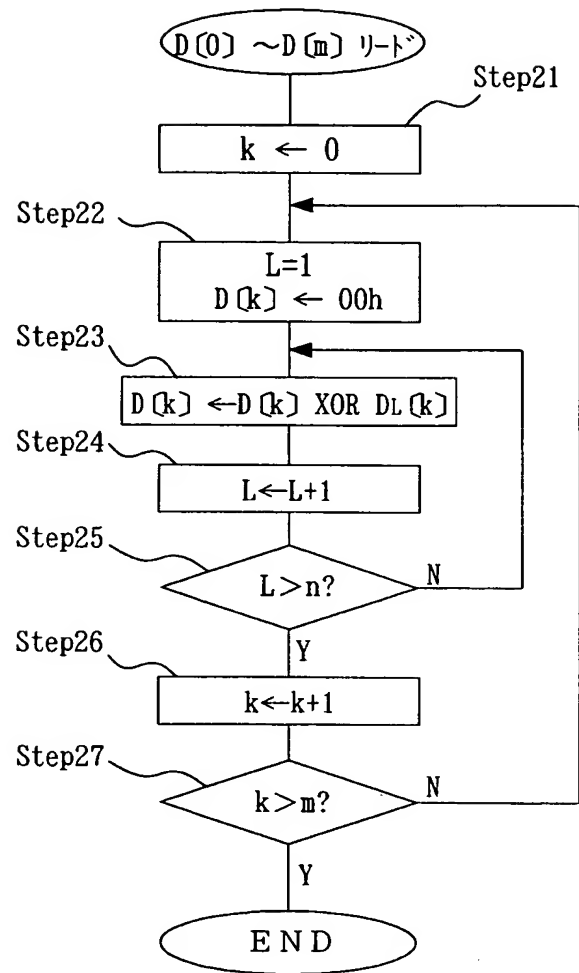
1 0 フラッシュメモリ、2 0 ワークメモリ、2 1 データ書込領域、X O R 排他的論理和。

【図 2】



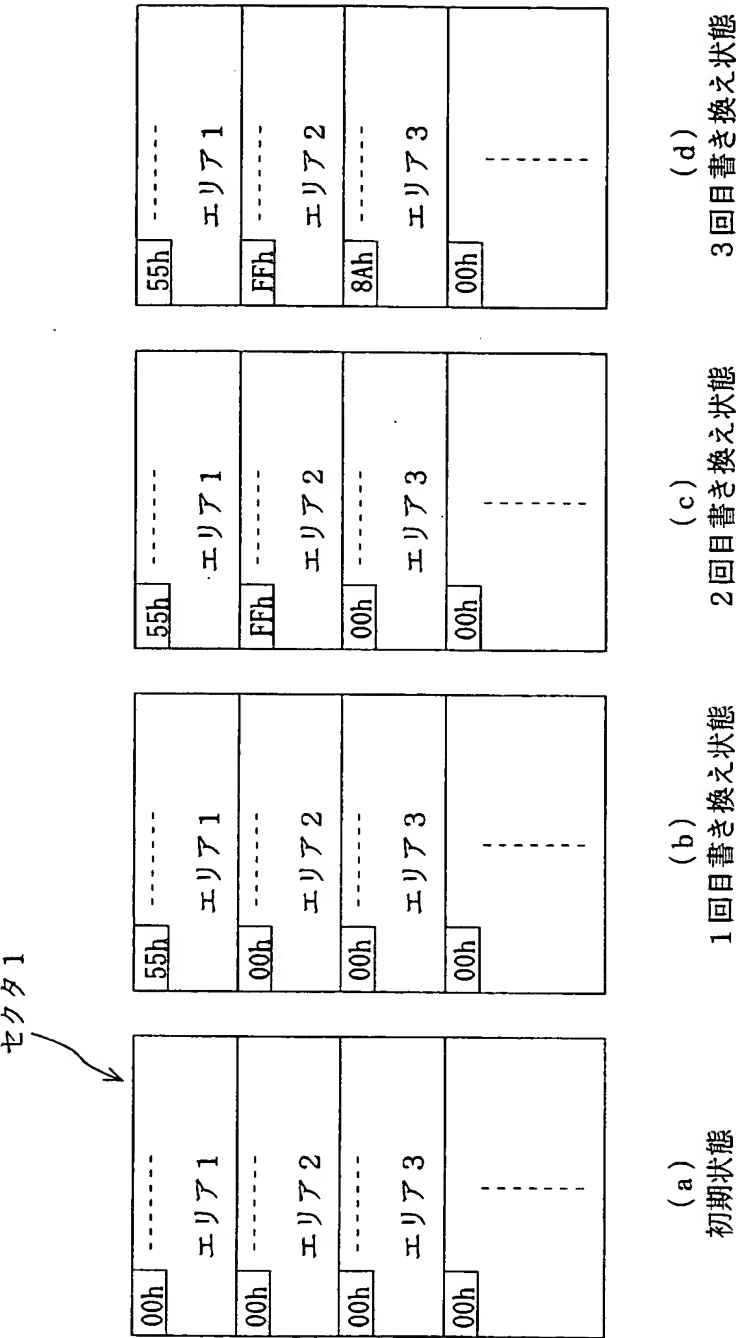
実施の形態 1 におけるデータ書き換え時のフローチャート

【図 3】



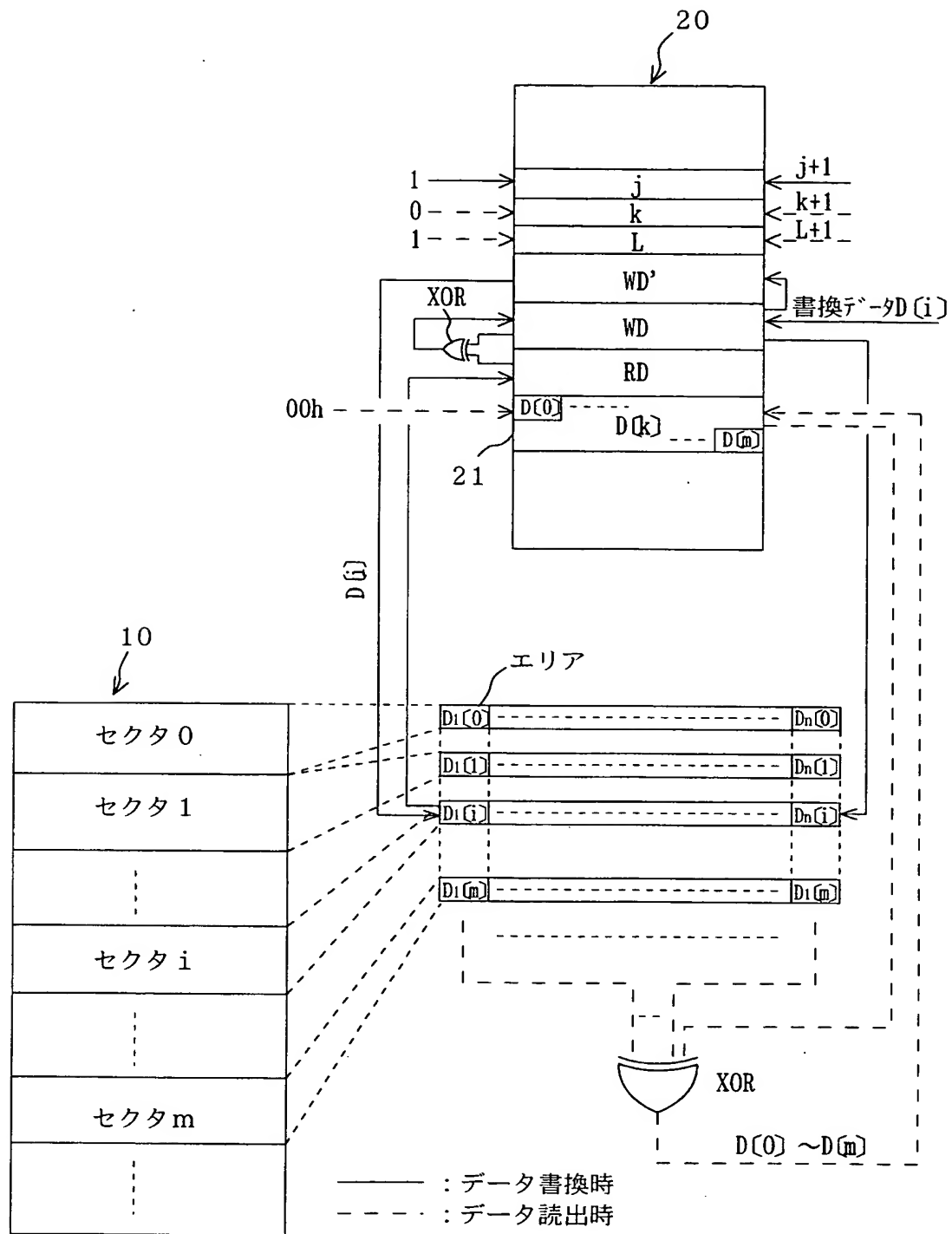
実施の形態 1 におけるデータ読み出し時のフローチャート

【図 4】



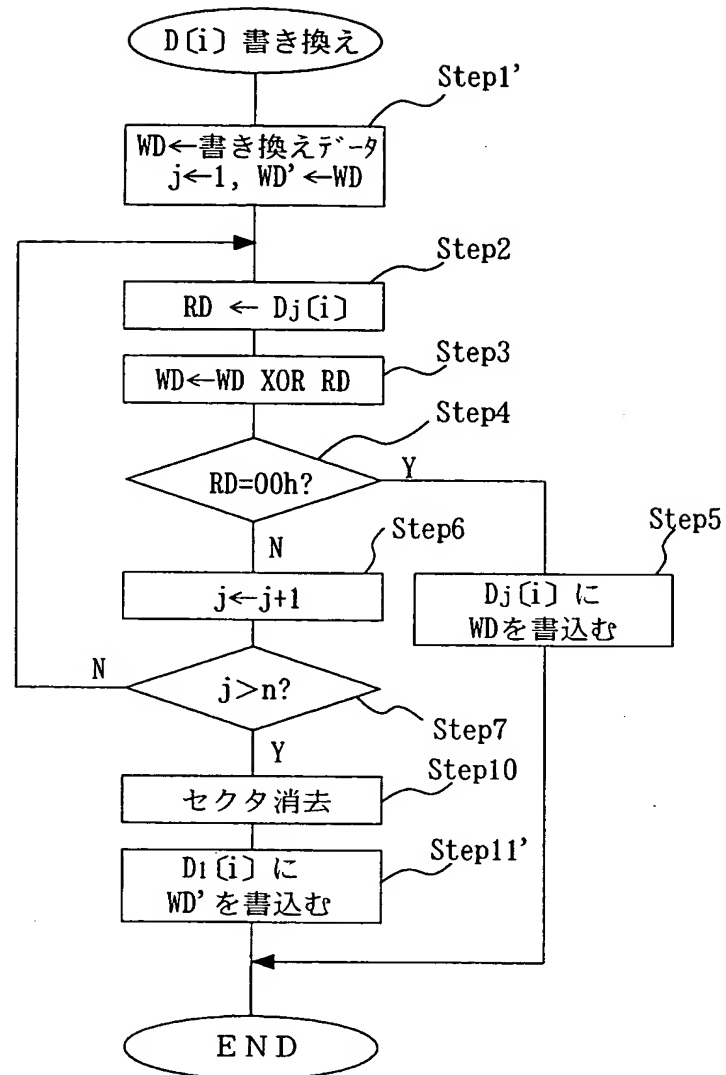
実施の形態 1 におけるデータ書き換え状態を示す図

【図 5】



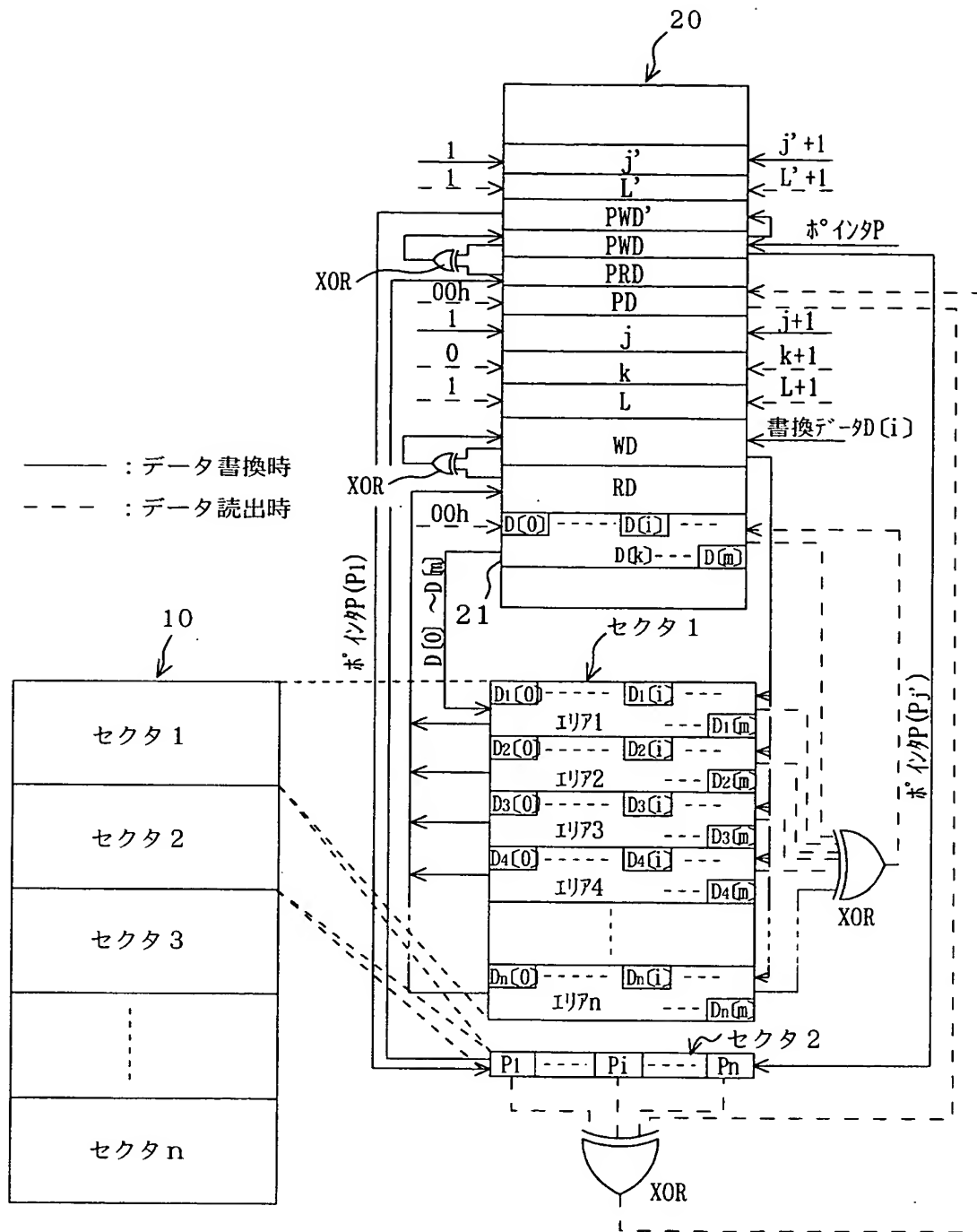
実施の形態 2 のデータ書換方法の説明図

【図 6】



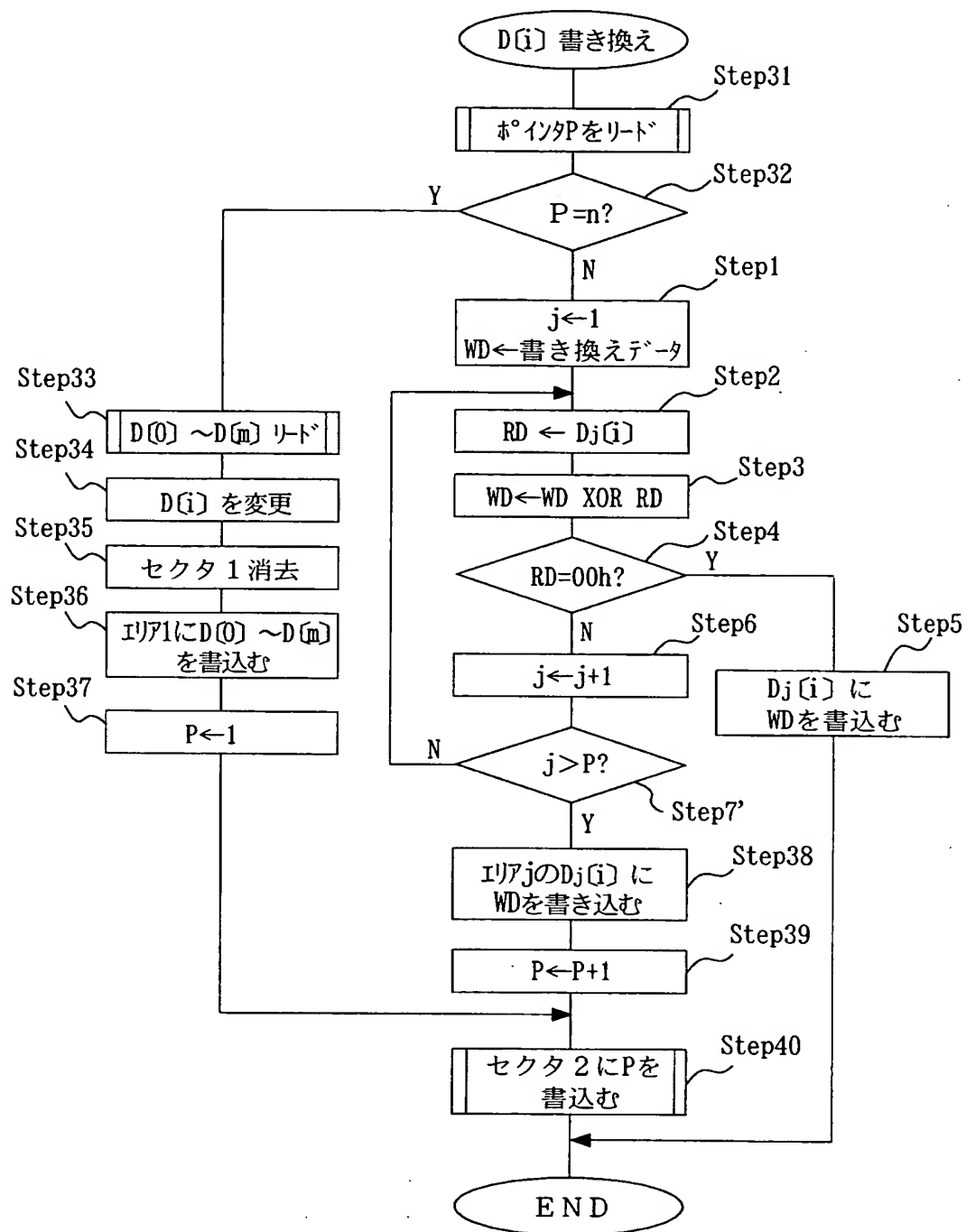
実施の形態 2 におけるデータ書き換え時のフローチャート

【図 7】



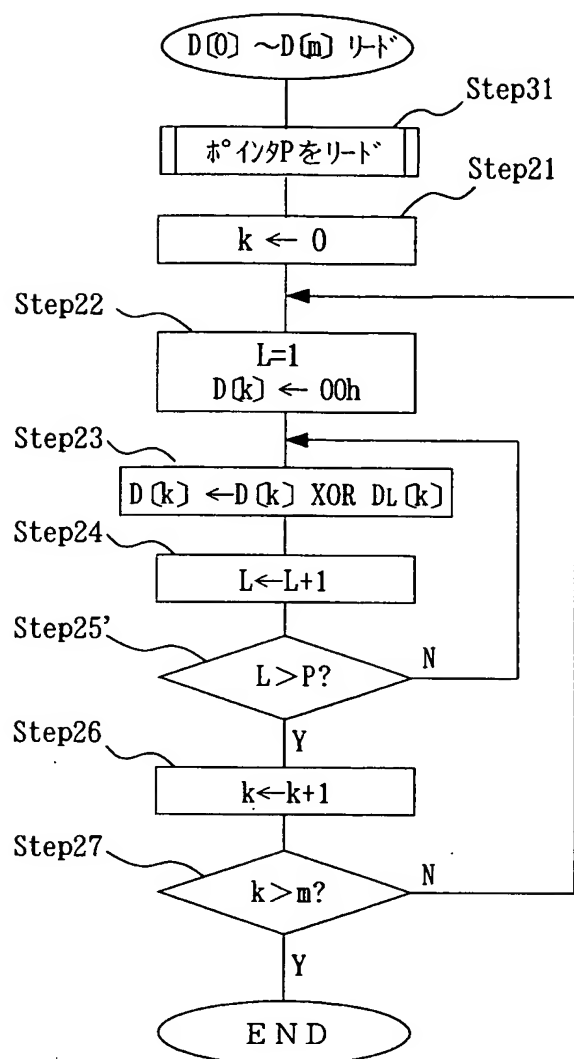
実施の形態 3 のデータ書換方法の説明図

【圖 8】



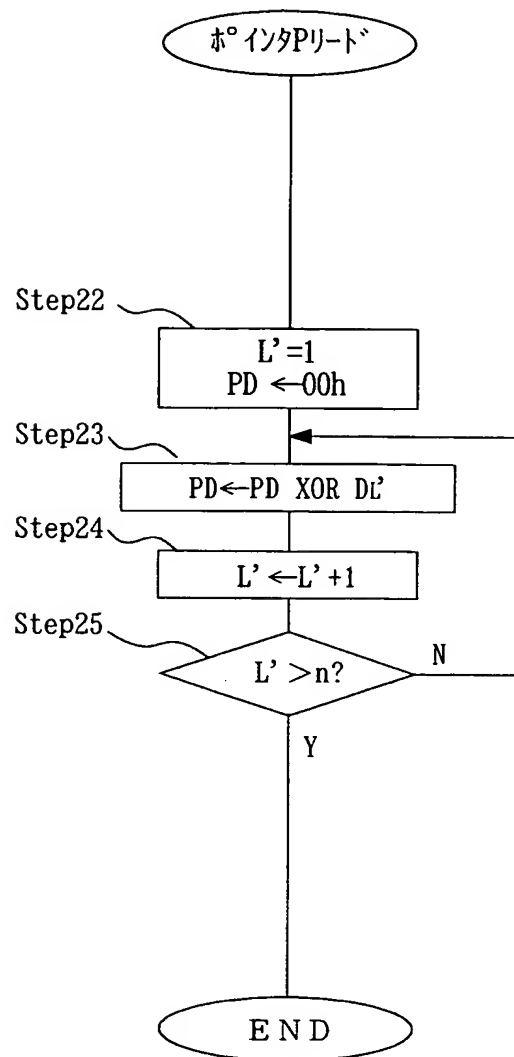
実施の形態3におけるデータ書き換え時のフローチャート

【図 9】



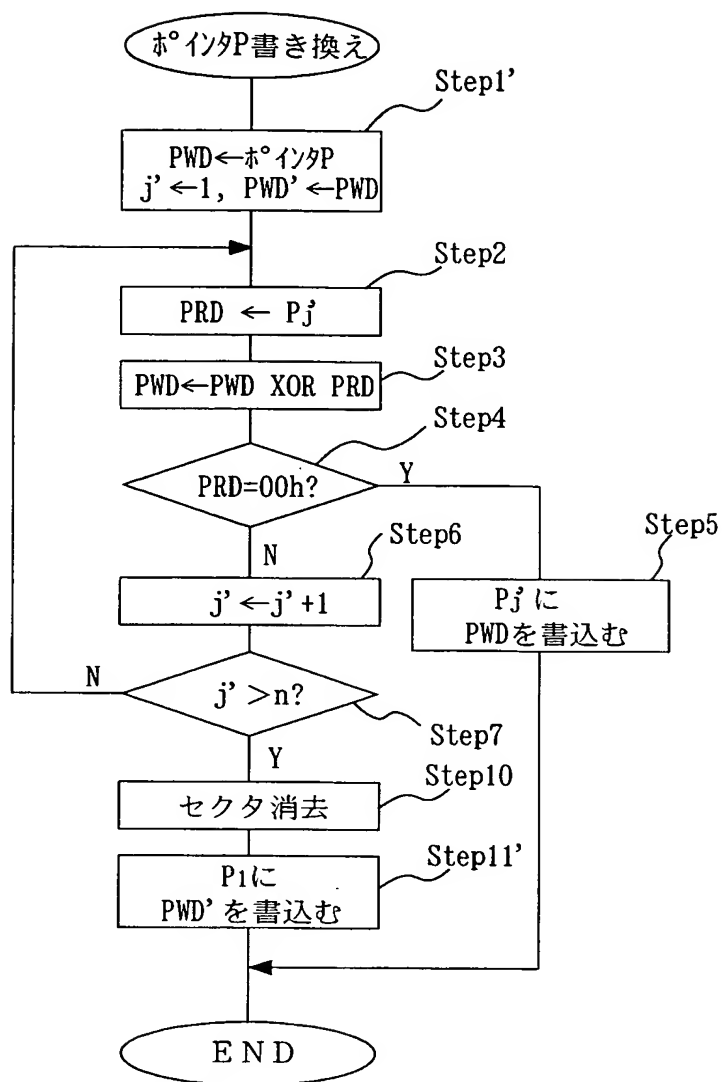
実施の形態 3 におけるデータ読み出し時のフローチャート

【図 10】



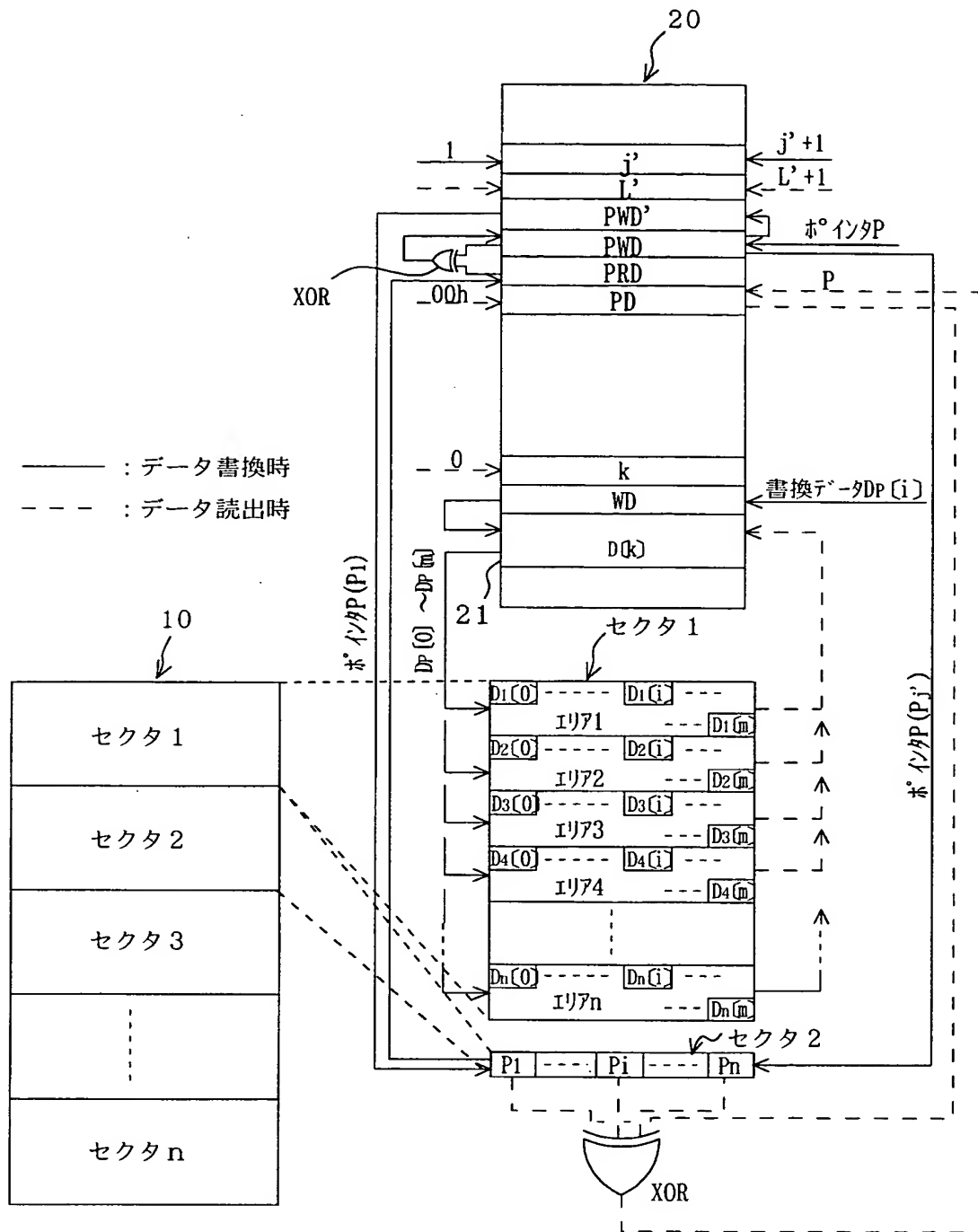
実施の形態 3 におけるポインタ P 読み出し時のフローチャート

【図 11】



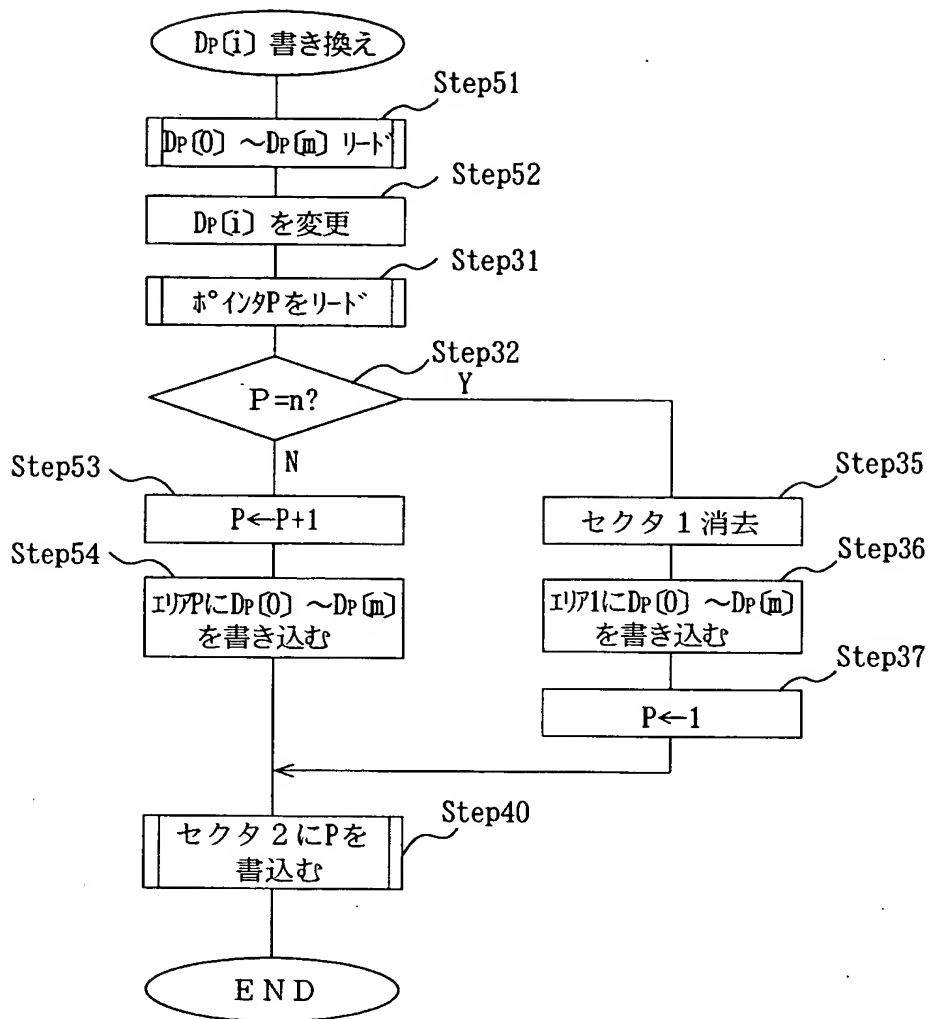
実施の形態3におけるポインタP書き換え時のフローチャート

【図 12】



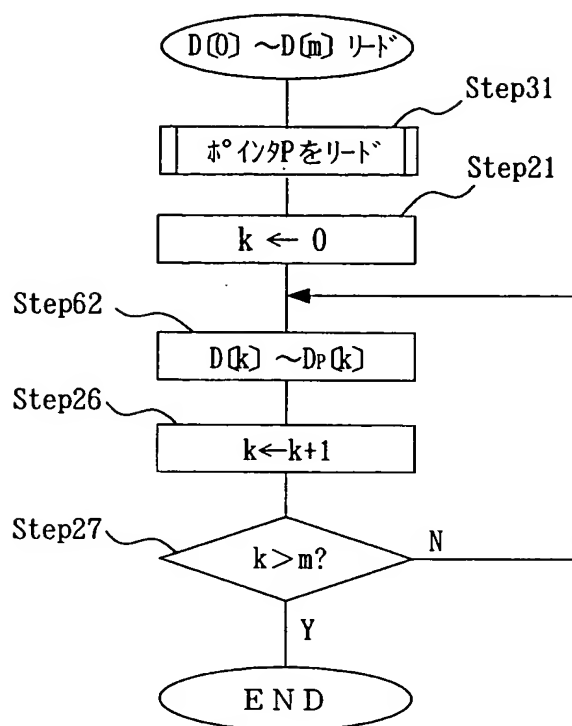
実施の形態 4 のデータ書換方法の説明図

【図 13】



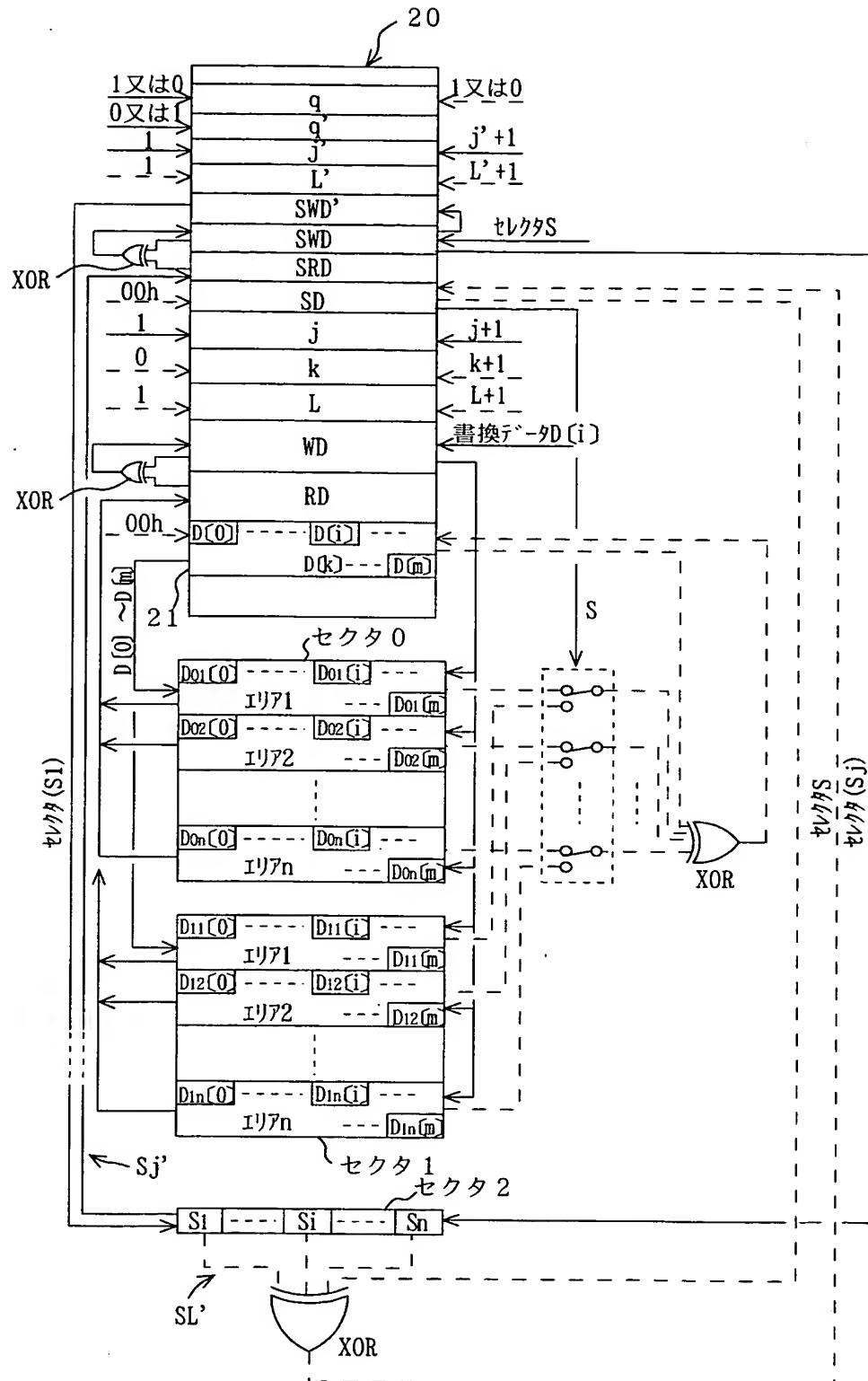
実施の形態 4 におけるデータ書き換え時のフローチャート

【図 14】



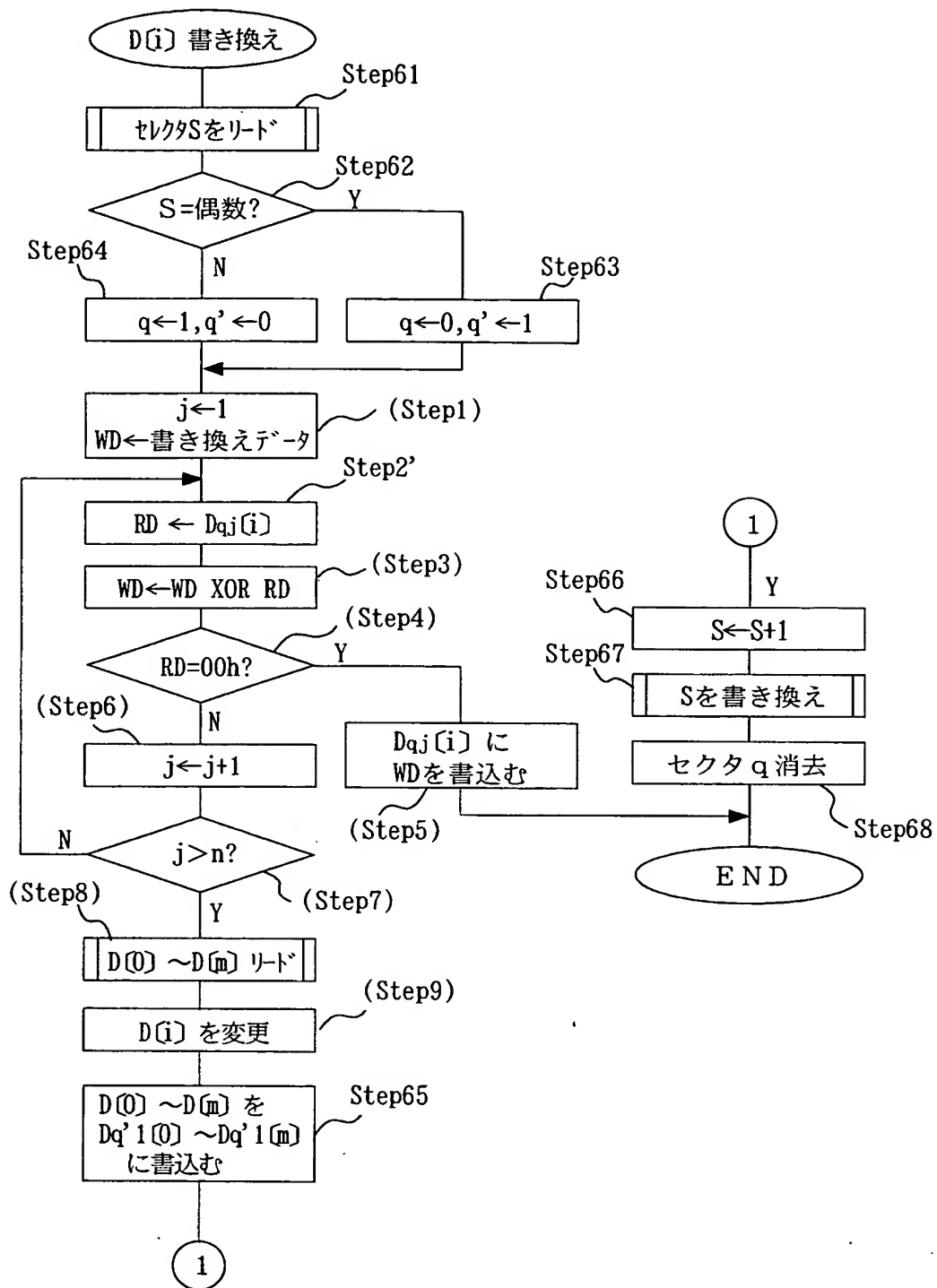
実施の形態 4 におけるデータ読み出し時のフローチャート

【図 15】



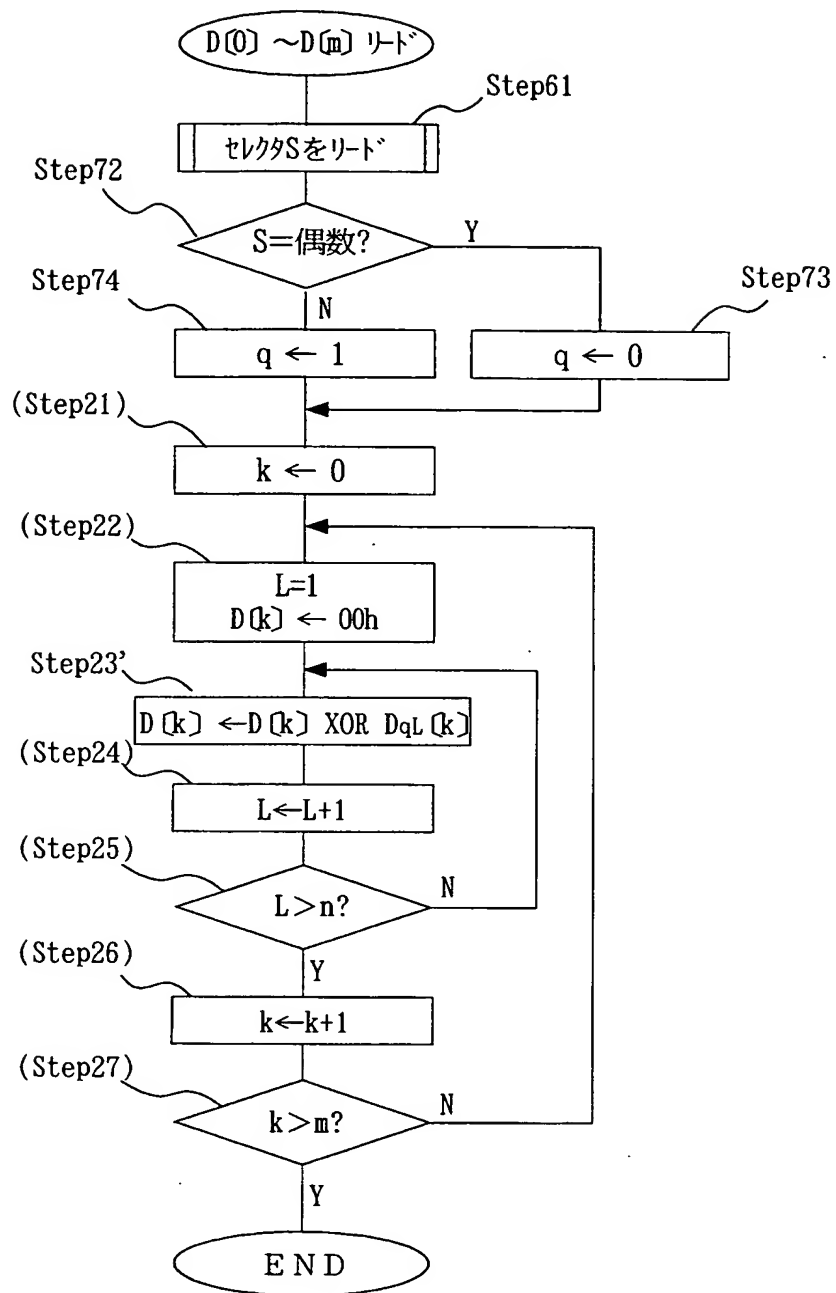
実施の形態 5 のデータ書換方法の説明図

【図 16】



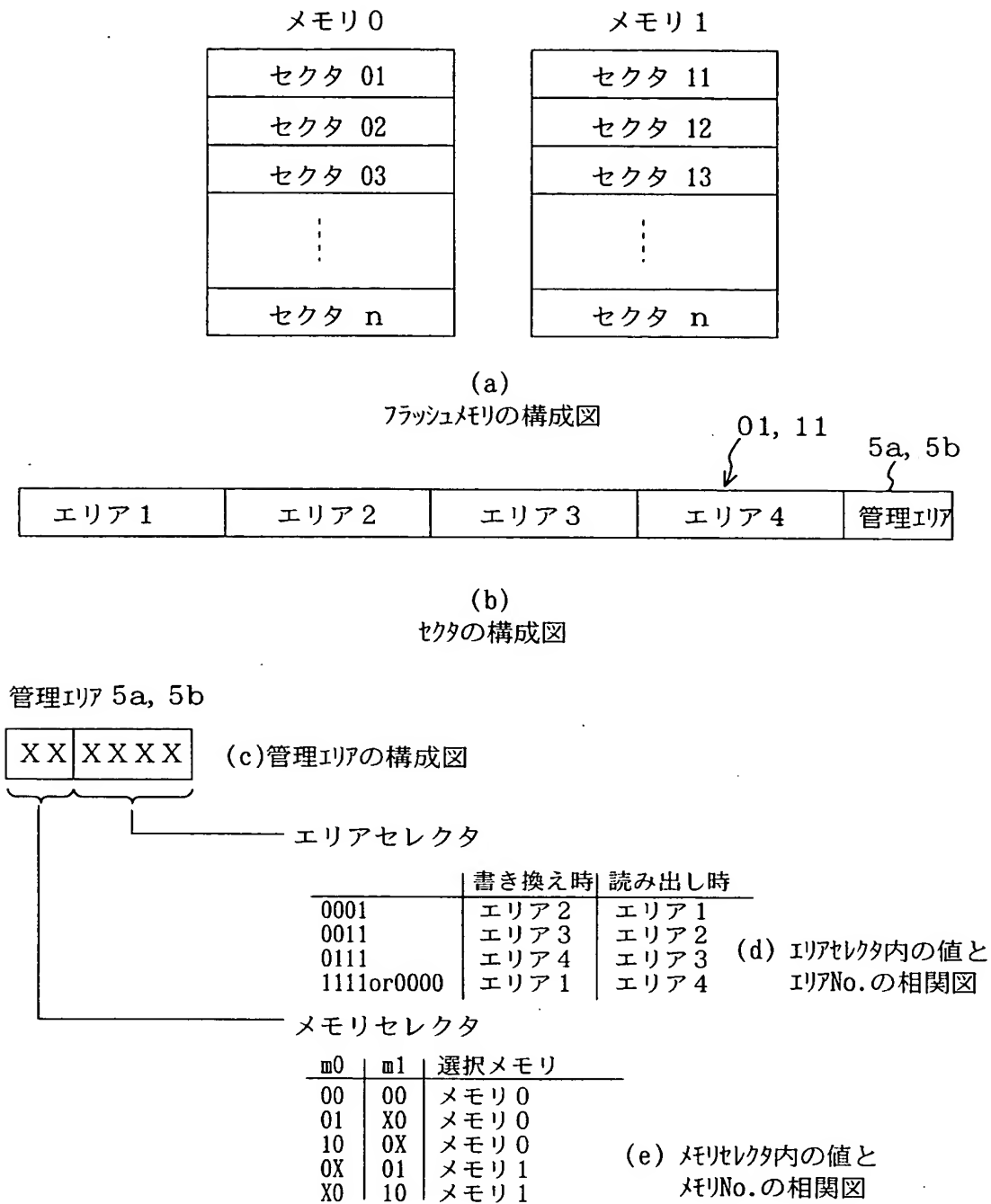
実施の形態 5 におけるデータ書き換え時のフローチャート

【図 17】



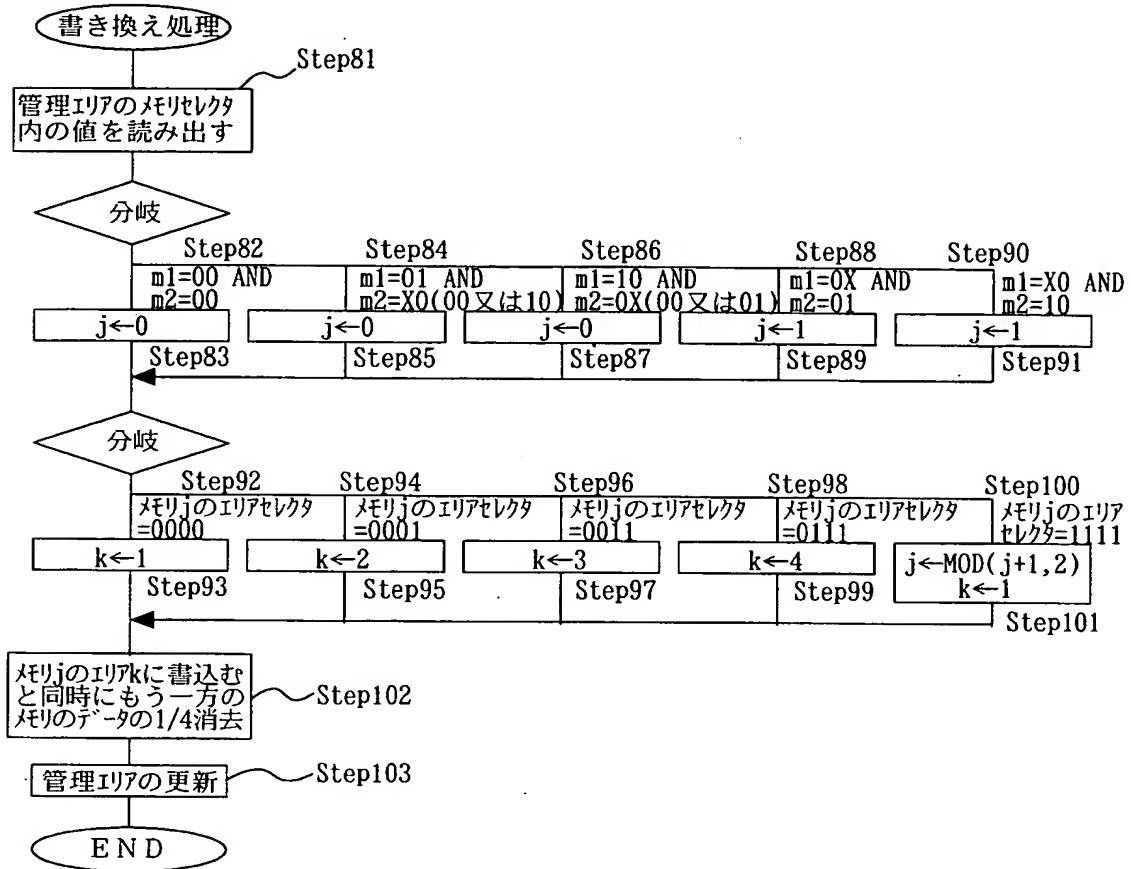
実施の形態5におけるデータ読み出し時のフローチャート

【図 18】



実施の形態 6 のデータ書換方法の説明図

【図 19】



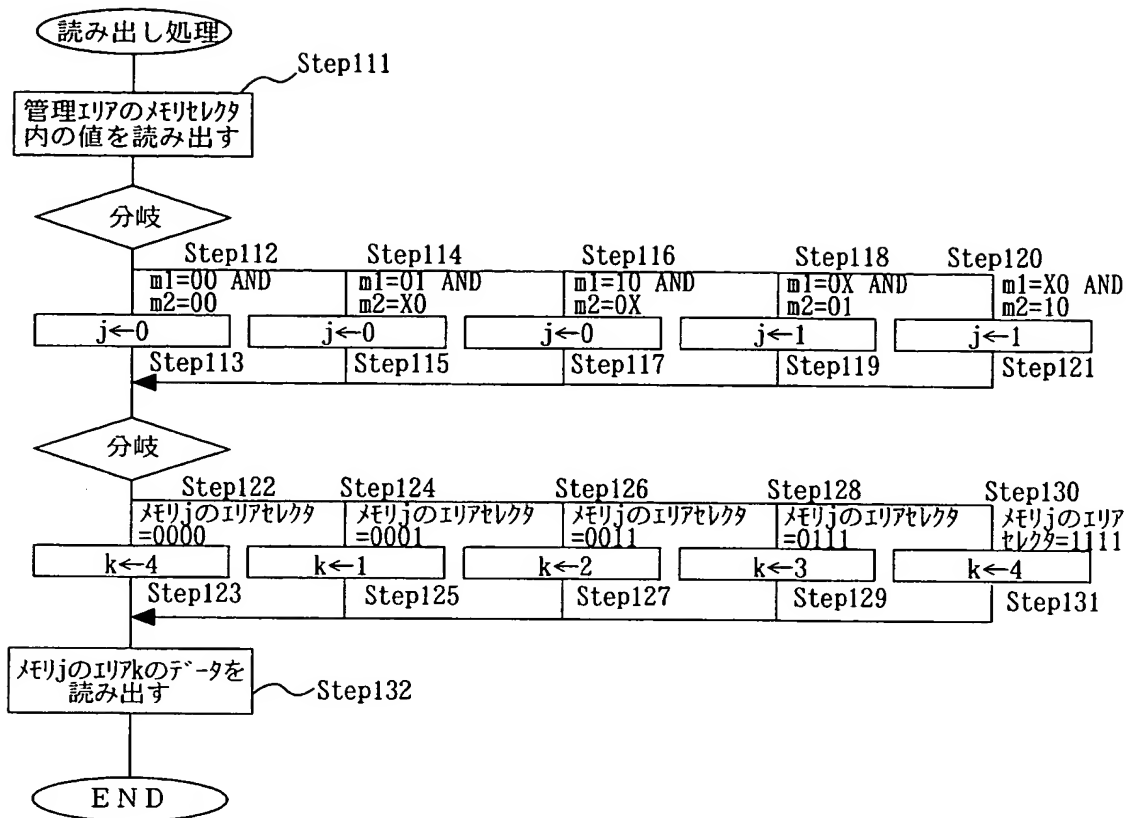
実施の形態 6 におけるデータ書き換え時のフローチャート

【図 20】

管理175a					メモリ0のセクタ01				メモリ1のセクタ11				管理175b
(a)	000000	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	000000	
(b)	010001	Data	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	000000	
(c)	010011	Data	Data	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	000000	
(d)	010111	Data	Data	Data	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	000000	
(e)	011111	Data	Data	Data	Data	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	000000	
(f)	0XXXXX	A11 XXh	A11 00h	A11 XXh	A11 XXh	Data	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	010001	
			⋮					⋮					
(g)	000000	A11 00h	A11 00h	A11 00h	A11 00h	Data	Data	Data	Data	Data	Data	011111	
(h)	100001	Data	A11 00h	A11 00h	A11 00h	A11 XXh	A11 XXh	A11 XXh	A11 XXh	A11 XXh	A11 XXh	0XXXXX	
			⋮					⋮					
(i)	101111	Data	Data	Data	Data	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	000000	
(j)	X0XXXX	A11 XXh	A11 XXh	A11 XXh	A11 XXh	Data	A11 00h	A11 00h	A11 00h	A11 00h	A11 00h	100001	
			⋮					⋮					
(k)	000000	A11 00h	A11 00h	A11 00h	A11 00h	Data	Data	Data	Data	Data	Data	101111	

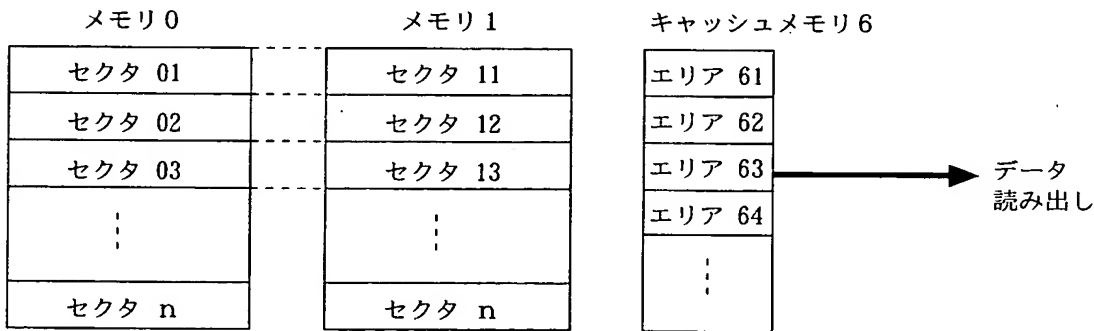
データ書き換え処理に基づくセクタの状態を示す図

【図 2 1】



実施の形態 6 におけるデータ読み出し時のフローチャート

【図 2 2】



実施の形態 7 の構成図

【書類名】 要約書**【要約】**

【課題】 高速でデータの書き換えができ、しかも書き換え回数の少ないフラッシュメモリのデータ書き換え方法を提供する。

【解決手段】 複数のエリア1～nに分割され、この各エリア1～nの同一位置にそれぞれデータが書き込まれるセクタ1を有するフラッシュメモリ10を備え、書換データを書き込む際、その書換データとエリア1の書換対象位置のデータとの排他的論理和をとり、さらに、この排他的論理和のデータとエリア2の前記位置のデータとの排他的論理和をとる処理を順次繰り返し、その過程で当該エリアの前記位置のデータが初期値であったとき、そのエリアの前記位置にそれまでに求めた排他的論理和のデータを書き込む。

【選択図】 図1

特願 2 0 0 3 - 2 7 4 7 3 7

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 0 2 9 5]

1. 変更年月日

1 9 9 0 年 8 月 2 2 日

[変更理由]

新規登録

住 所

東京都港区虎ノ門1丁目7番12号

氏 名

沖電気工業株式会社